American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16

Please solve the following exercises and submit **BEFORE 11:55 p.m. of Friday 18 September.**

Exercise 1 (10 points)

Which of these are propositions? What are the truth values of those that are propositions?

a) The north pole is hot.
Proposition - False
b) When is the closest holiday?
Not a proposition
c) $z^2 > -1$, for all z $\in C$
Proposition - True
d) Do not pass! Just go!
Not a proposition
e) In CMPS 211, there is no raise
Proposition. True if there's no raise, false otherwise.

Exercise 2 (10 points)

Let f, e, and h be the propositions

- f : You eat healthy food
- e : You exercise regularly
- h : You are in good health

Write these propositions using f, e, and h and logical connectives (including negations).

a) If you aren't in good health, then you either don't eat healthy food, or you don't exercise regularly.
$\neg h \rightarrow (\neg f \vee \neg e)$
b) You neither eat healthy food, nor you exercise regularly, and you aren't in good health
$\neg f \wedge \neg e \wedge \neg h$
c) To exercise regularly, it is necessary that you eat healthy food
$e \rightarrow f$
d) Exercising regularly is sufficient for being in good health
$e \rightarrow h$
e) If you aren't in good health, then you don't eat healthy food, and if you don't eat

healthy food, then you can't exercise regularly.

$(\neg h \rightarrow \neg f) \wedge (\neg f \rightarrow \neg e)$

Exercise 3 (10 points)

Determine whether these bi-conditionals are true or false.

f) $-\infty > +\infty$ if and only if $\frac{0}{0}$ is defined
true

g) 5*5=25 if and only if 1-1=2.
false

h) $(p \vee \neg p) \leftrightarrow$ Bananas are yellow;
true

Exercise 4 (10 points)

Let p and q be the propositions

- d: I do my assignments on my own
- p: I pass my courses

Express each of these propositions as an English sentence.

i) $p \rightarrow d$
if I pass my courses then I've done my assignments on my own

j) $\neg d \wedge \neg p$
I don't do my assignments on my own and I don't pass my courses

k) $\neg d \vee (d \wedge p)$
I either don't do my assignment on my own, or I do my assignments on my own and I pass my courses

Exercise 5 (10 points)

For each of these sentences, determine whether an inclusive or, or an exclusive or, is intended. Explain your answer.

a) To enter the country you need a passport or a voter registration card.
inclusive

b) You can pick Section 1 or Section 2
exclusive

c) You either own a Samsung or an iPhone
inclusive

d) You either get a full grade or you have some mistake
   exclusive

Exercise 6 (10 points)

State the converse, contrapositive, and inverse of each of these conditional statements.

a) If you solve it fast, then it's wrong
   p: solve fast, q: its wrong
   converse: if it's wrong, then you'll solve it fast
   contrapositive: if its not wrong, then you won't solve it fast
   inverse: if you aren't solving it fast, then it's not wrong
b) To stay healthy, it's sufficient to drink milk
   p: drink milk, q: stay healthy
   converse: if you are healthy then you drink milk
   contrapositive: if you aren't healthy then you don't drink milk
   inverse: if you don't drink milk then you aren't healthy
c) When I wake up late, it is necessary that I have a full day headache
   p: I wake up late, q: I have a full day headache
   converse: If I have a full day headache, then I woke up late
   contrapositive: If I didn't have a full day headache, then I didn't wake up late
   inverse: if I don't wake up late, I won't have a full day headache

Exercise 7 (10 points)

Construct a truth table for each of these compound propositions.

a) ¬p→¬q
b) p↔(¬p∧ q)
c) (¬p∨¬q)→p
d) (¬p→¬q)↔(q→p)
e) (p↔q) XOR (p↔¬q)

p	q	¬p	¬q	**¬p→¬q**	¬p∧ q	**p↔(¬p∧ q)**	(¬p∨¬q)	**(¬p∨¬q)→p**	(q→p)	**(¬p→¬q) ↔(q→p)**	(p↔q)	(p↔¬q)	**(p↔q) XOR (p↔¬q)**
0	0	1	1	**1**	0	**1**	1	**0**	1	**1**	1	0	**1**
0	1	1	0	**0**	1	**0**	1	**0**	0	**1**	0	1	**1**
1	0	0	1	**1**	0	**0**	1	**1**	1	**1**	0	1	**1**
1	1	0	0	**1**	0	**0**	0	**1**	1	**1**	1	0	**1**

Exercise 8 (10 points)

What is the value of x after each of these statements is encountered in a computer program, if x = -1 before the statement is reached?

a) if x - 3 < 0 then x := (x+1)/2 +1
   if -1 - 3 < 0 then x := (-1+1)/2 +1, true → x = 1
b) if (2x +2 =3) OR (3x +1 = 3) then x := x*2
   if (2(-1) +2 =3) OR (3(-1) +1 = 3) then x := -1*2, false → x = -1
c) if (x + 3 = 2) AND (3x +4 = -1) then x := x-2
   if (-1 + 3 = 2) AND (3(-1) +4 = -1) then x := x-2, false → x = -1
d) if (x > x) XOR (x < x) then x := x+2
   if (-1 > -1) XOR (-1 < -1) then x := -1+2, false → x = -1
e) if x < 2 then x := x mod 2 –mod is the modulus function
   if -1 < 2 then x := -1 mod 2, true → x = 1

Exercise 9 (20 points)

Fuzzy logic is used in artificial intelligence. In fuzzy logic, a proposition has a truth value that is a number between 0 and 1, inclusive. A proposition with a truth value of 0 is false and one with a truth value of 1 is true. Truth values that are between 0 and 1 indicate varying degrees of truth.

For instance, the truth value 0.3 can be assigned to the statement "Tarek is *sick*" because Tarek is *sick* slightly less than half the time, and the truth value 0.9 can be assigned to the statement "Reem is *hungry*" because Reem is *hungry* most of the time. **Use these truth values to solve the following exercises**.

a) The truth value of the negation of a proposition in fuzzy logic is 1 minus the truth value of the proposition. What are the truth values of the statements "Tarek is not *sick*", and "Reem is not *hungry*?"
   Tarek is not *sick* = 1-0.3 = 0.7
   Reem is not *hungry* = 1-0.9 = 0.1

b) The truth value of the conjunction of two propositions in fuzzy logic is the minimum of the truth values of the two propositions. What are the truth values of the statements "Tarek is *sick* and Reem is *hungry*" and "Tarek is not *sick* but Reem is *hungry*?"
   Tarek is *sick* and Reem is *hungry* = min(0.3, 0.9) = 0.3
   Tarek is not *sick* but Reem is *hungry* = min(0.7, 0.9) = 0.7

c) The truth value of the disjunction of two propositions in fuzzy logic is the maximum of the truth values of the two propositions. What are the truth values of the statements "Tarek is *sick*, or Reem is not *hungry*" and "Tarek is not *sick*, or

Reem is not *hungry*?"

Tarek is *sick*, or Reem is not *hungry* = max(0.3, 0.1) = 0.3

Tarek is not *sick*, or Reem is not *hungry* = max(0.7, 0.1) = 0.7

American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16
Assignment 2 Solution

Please solve the following exercises and submit **BEFORE 11:55 pm of Wednesday 7th, October.** Submit on Moodle.

Exercise 1 (10 points)

Determine whether each of these compound propositions following is satisfiable or not. When satisfiable, give the satisfying assignments for the variables, build truth table when it's not satifiable.

a) (p ∨ q ∨ r ∨ s) ∧ (¬p ∨ q ∨ r ∨ ¬s) ∧ (¬p ∨ ¬q ∨ ¬r ∨ s) ∧ (p ∨ ¬q ∨ r ∨ ¬s)
Satisfiable for p = true, q = true, r = false
b) (p ∨ q ∨ ¬s ∨ ¬r) ∧ (p ∨ ¬q ∨ s ∨ r) ∧ (p ∨ q ∨ s ∨ r) ∧ (¬p ∨ q ∨ s ∨ r) ∧ (p ∨ q ∨ r ∨ s) ∧ (¬p ∨ ¬q ∨ ¬r ∨ ¬s)
Satisfiable for p = true, q = true, r = false
c) (p ∨ q) ∧ (¬p ∨ ¬q) ∧ (p ∨ ¬q ∨ r)
Satisfiable for p = true, q = false

Exercise 2 (10 points)

Determine whether the following are tautologies **without** using truth tables

a) p ∧q → p ∨ q
p ∧q → p ∨ q
⇔ ¬(p ∧q) ∨ (p ∨ q)
⇔ (¬p ∨ ¬q) ∨ (p ∨ q)
⇔ (¬p ∨ p) ∨ (¬q ∨ q)
⇔ T ∨ T
→ Tautology
b) [(q → p) ∧ (r ∧ p) ∧ (p → q)] → p
[(q → p) ∧ (r ∧ p) ∧ (p → q)] → p
⇔ [(¬q ∨ p) ∧ (r ∧ p) ∧ (¬p ∨ q)] → p
⇔ ¬[(¬q ∨ p) ∧ (r ∧ p) ∧ (¬p ∨ q)] ∨ p
⇔ [¬ (¬q ∨ p) ∨ ¬ (r ∧ p) ∨ ¬ (¬p ∨ q)] ∨ p
⇔ (q ∧ ¬p) ∨ (¬r ∨ ¬p) ∨ (p ∧ ¬q) ∨ p
⇔ (q ∧ ¬p) ∨ (p ∧ ¬q) ∨ ¬r ∨ (¬p ∨ p)
⇔ (q ∧ ¬p) ∨ (p ∧ ¬q) ∨ ¬r ∨ T
→ Tautology
c) (p → q) ↔ [¬p ∨ (p ∧q)]
(p → q) ↔ [¬p ∨ (p ∧q)]
⇔ (¬p ∨ q) ↔ [(¬p ∨ p) ∧ (¬p ∨q)]
⇔ (¬p ∨ q) ↔ [T ∧ (¬p ∨q)]

⇔ (¬p ∨ q) ↔ (¬p ∨ q)
→ Tautology

Exercise 3 (15 points)

Consider the logical operations *NAND*. The proposition *p NAND q* is false when p and q are both true, and true otherwise. The propositions p *NAND* q is denoted by p | q.

a) Construct a truth table for the logical operator *NAND*.

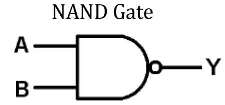
p	q	p | q
T	T	F
T	F	T
F	T	T
F	F	T

b) Show that p | q is logically equivalent to ¬(p∧q).

p	q	p∧q	¬(p∧q)	p | q	¬(p∧q) ↔ p | q
T	T	T	F	F	T
T	F	F	T	T	T
F	T	F	T	T	T
F	F	F	T	T	T

c) NAND gates are universal gates, meaning that you can create any other gate using NAND gates only, combined in different ways. Using NAND gates only draw circuits that can operate in equivalence to:

NAND Gate



i. Not gate
¬p ⇔ p | p

ii. AND gate
(p∧q)
⇔ ¬ (p | q)
⇔ (p | q) | (p | q)

iii. OR gate
(p ∨ q)
⇔ ¬ (¬p ∧¬q)
⇔ ¬[(p | p) ∧ (q | q)]
⇔ ¬[{(p | p) | (q | q)} | {(p | p) | (q | q)}]
⇔ ¬[{(p | p) | (q | q)} | {(p | p) | (q | q)}]
⇔ [{(p | p) | (q | q)} | {(p | p) | (q | q)}] | [{(p | p) | (q | q)} | {(p | p) | (q | q)}]
Also equivalent to ¬ p | ¬q

iv. NOR gate (p NOR q is true when both p and q are false, false otherwise)
$(p \downarrow q)$
$\Leftrightarrow \neg(p \vee q)$
$\Leftrightarrow (\neg p \wedge \neg q)$
$\Leftrightarrow (\neg p \mid \neg q) \mid (\neg p \mid \neg q)$
$\Leftrightarrow ((p \mid p) \mid (q \mid q)) \mid ((p \mid p) \mid (q \mid q))$
Also equivalent to $\neg(\neg p \mid \neg q)$

Exercise 4 (10 points)

Determine whether $\forall x(P(x) \leftrightarrow Q(x))$ and $\forall x\, P(x) \leftrightarrow \forall x Q(x)$ are logically equivalent. Justify your answer.

No, they aren't logically equivalent.

Assume $P(x) = x < 0$, and $Q(x) = x \geq 0$, and the domain of x is all integers;

- $\forall x\, P(x)$ is false, and $\forall x\, Q(x)$is also false, then $\forall x\, P(x) \leftrightarrow \forall x Q(x)$ is true
- but P(1) is false and Q(1) is true, so $P(1) \leftrightarrow Q(1)$ is false, then $\forall x(P(x) \leftrightarrow Q(x))$ is false

Then $\forall x\, P(x) \leftrightarrow \forall x Q(x)$ is not logically equivalent to $\forall x(P(x) \leftrightarrow Q(x))$

Exercise 5 (10 points)

Translate the given statement into propositional logic using the propositions provided:

You can upgrade your operating system only if you have a 32-bit processor running at 1 GHz or faster, at least 1 GB RAM, and 16 GB free hard disk space, or a 64-bit processor running at 2 GHz or faster, at least 2 GB RAM, and at least 32 GB free hard disk space.

Express you answer in terms of:

- *u*: "You can upgrade your operating system"
- *b*32: "You have a 32-bit processor"
- *b*64: "You have a 64-bit processor"
- *g*1: "Your processor runs at 1 GHz or faster"
- *g*2: "Your processor runs at 2 GHz or faster"
- *r*1: "Your processor has at least 1 GB RAM"
- *r*2: "Your processor has at least 2 GB RAM"
- *h*16: "You have at least 16 GB free hard disk space"
- *h*32: "You have at least 32 GB free hard disk space"

$u \rightarrow (b32 \wedge g1 \wedge r1 \wedge h16) \vee (b64 \wedge g2 \wedge r2 \wedge g32)$

Exercise 6 (10 points)

Let

- *P(x)* = "*x* is a clear explanation",
- *Q(x)* = "*x* is satisfactory"
- *R(x)* ="*x* is an excuse"

where the domain for *x* consists of all English text.

Express each of these statements using quantifiers, logical connectives, and *P(x)*, *Q(x)*, and *R(x)*. Then express their negations in English and using quantifiers.

a) All clear explanations are satisfactory.
   $\forall x\ (P(x) \rightarrow Q(x))$
b) Some excuses are unsatisfactory.
   $\exists x\ R(x) \wedge \neg Q(x)$
c) Some excuses are not clear explanations.
   $\exists x\ \neg P(x) \wedge R(x)$
d) Does (c) follow from (a) and (b)?
   Yes it follows. Since being unsatisfactory is guaranteed in (b), then being unclear is deduced from (a)

1.	$\exists x\ R(x) \wedge \neg Q(x)$	Premise
2.	$R(c) \wedge \neg Q(c)$	Existential Instantiation from 1
3.	$\neg Q(c)$	Simplification from 2
4.	$\forall x\ (P(x) \rightarrow Q(x))$	Premise
5.	$\forall x\ (\neg Q(x) \rightarrow \neg P(x))$	Contraposition from 4
6.	$\neg Q(c) \rightarrow \neg P(c)$	Universal Instantiation from 5
7.	$\neg P(c)$	Modus Pollens from 3 and 6
8.	$R(c)$	Simplification from 2
9.	$R(c) \wedge \neg P(c)$	Conjugation from 7 and 8
10.	$\exists x\ \neg P(x) \wedge R(x)$	Existential Generalization from 9

Exercise 7 (10 points)

Suppose the propositional function $P(x) = x^2 - 5 > 12 \rightarrow x^3 + 9 > -10$, and the domain that consists of {−10, -4, 0, 4}. Express these statements without using quantifiers, instead using only negations, disjunctions, and conjunctions, and evaluate each statement.

a) ∃xP(x)
P(-10) ∨ P(-4) ∨ P(0) ∨ P(4)
⇔ (95 > 12 → -991 > -10) ∨ (11 > 12 → -55>-10) ∨ …∨
⇔(T → F) ∨ (F → F) ∨..∨
⇔ F ∨ T ∨ … ∨…
⇔ T

b) ∀xP(x)
P(-10) ∧ P(-4) ∧ P(0) ∧ P(4)
False, since P(-4) is false

c) ∀x((x≠10)→P(x))
x ≠ 10 is always true in our domain, then ∀x((x≠10)→P(x)) ↔ ∀xP(x) in part c, then false

d) ∃x(¬P(x)) ∧ ∀x((x>4)→P(x))
We know that P(-10) is only false, and all the rest are true; since we need ¬P(x), then we can plug in -10 for x in ∃x(¬P(x)), instead of enumerating all the domain and combine with disjunctions
→ P(-10) ∧ [(-10 > 4 →P(-10)) ∧ (-4 > 4 →P(-4)) ∧ (0 > 4 →P(0)) ∧ (4 > 4 →P(4))
⇔T ∧ (F → T ∧ F → F ∧ F → F ∧ F → F)
⇔ T ∧ T ∧ T ∧ T ∧ T
⇔ T

Exercise 8 (15 points)

Translate in two ways each of these statements into logical expressions using predicates, quantifiers, and logical connectives. First, let the domain consist of all transportations tools (Cars, Motorcycle, Buses, Planes, Trains, Horses, etc..) and the second be cars.

Let C(x) = "x is a car", where x belongs to all transportation tools
Let O(p, x) = "person p owns object x", where domain of p belong to all people

a) Some cars run on solar power
Let S(x) = "x runs on solar power" where x belongs to all transportation tools
1. ∃x C(x) ∧ S(x)
2. ∃x S(x)

b) All people own at least one car
1. ∀p ∃x (C(x) ∧ O(p,x))
2. ∀p ∃x O(p,x)

c) Some people own 2 (or more) cars
   1. ∃p ∃x ∃y x≠y ∧ C(x) ∧ C(y) ∧ O(p,x) ∧ O(p,y)
   2. ∃p ∃x ∃y x≠y ∧ O(p,x) ∧ O(p,y)

d) Some people own no car.
   1. ∃p ∀x (¬C(x) ∨ ¬O(p,x))
   2. ∃p ∀x ¬O(p,x)

e) Some cars are faster than all non-cars (*other transport tools)*
   Let F(x,y) = "x is faster than y"
   1. ∃x ∀y (C(x) ∧ ¬C(y) → Faster(x, y))
   2. Can't be expressed since the domain only contain cars [unless we create a new domain for non-cars]…

f) Exactly 2 cars crashed last night
   Let Crashed(x) = "x crashed last night"
   1. *2 more:* ∃x ∃y ∀z (C(x) ∧ C(y) ∧ Crashed(x) ∧ Crashed(y) ∧ x≠y)
      Exactly 2: ∃x ∃y ∀z [C(x) ∧ C(y) ∧ Crashed(x) ∧ Crashed(y) ∧ x≠y ∧ (x = z ∨ y = z ∨ ¬C(z) ∨ ¬Crashed(z))]
   2. ∃x ∃y ∀z [Crashed(x) ∧ Crashed(y) ∧ x≠y ∧ (x = z ∨ y = z ∨ ¬Crashed(z))]

Exercise 9 (10 points)

Express these propositions and their negations using quantifiers, and in English.

a) There is a soccer player who didn't score any goal.
   Let S(x): "player x score at least a goal"
   Original: ∃x ¬ S (x)
   English Negation: All score players scored at least one goal
   Propositional Negation: ∀x S(x)

b) Every professor taught all courses in his department
   Let
   - T(p, c) = "Prof. p taught course c",
   - O(c, d) = "Course c is offered in dept. d"
   - M(p, d) = "Prof p is a member of dept d"

   where domain of p is professors, and domain of c is courses, and d is departments
   Original: ∀p ∀d ∀c [O(c,d) ∧ M(p,d) → T(p,c)]
   English Negation: Some professors didn't teach a course in their department
   Propositional Negation: ∃p ∃d ∃c [O(c, d) ∧ M(p, d) ∧ ¬T(p, c)]
   For simplicity you may assume domain of c is courses in department d which the professor is a member of, and then remove the departments domains

c) Some taxi drivers have passed through at least one street in all area of Lebanon.
Let P(t, s, a) = "Taxi driver *t* passed through street *s* in area *a* in Lebanon", where the domains are obvious
Original: ∃p ∀a ∃s P(t, s, a)
English Negation: All taxi drivers have a street in some area in which they didn't pass through
Propositional Negation: ∀p ∃a ∀s ¬P(t, s, a)

d) Each lab has a computer that was never used by any student.
Let U(s, c ,l) = "Student s used computer c in lab l", where the domains are also obvious
Original: ∀l ∃c ∀s ¬U(s, c, l)
English Negation: All computers in some labs where used by students *OR* Some labs has no computers which were never used by students
Propositional Negation: ∃l ∀c ∃s U(s, c, l)

Exercise 10 (20 points)

Let F(x, y) be the statement "person x is a Facebook friend of person y", and K(x, y) "person x knows person y", S(x, z): "x is a student at university y", where the domain x and y is people, and that of z is universities. Express each of those statements and their negations in English/using quantifiers.

Note that F and K are commutative functions, meaning that F(x, y) = F(y, x) and K(x, y) = K(y, x)

a) All People who are Facebook friends with Joe know Joe
∀x F(x, Joe) → K(x, Joe)

b) Some people know everyone but isn't Facebook friend with anyone
∃x ∀y K(x, y) ∧ ¬F(x, y)

c) All university students are Facebook friend with each other
Let U(x) = "x is a university student"
∀x∀y [U(x) ∧ U(y) → F(x, y)]
OR if you don't need to create U(x):
∀x ∀z1 ∀y ∀z2 [(S(x, z1) ∧ S(y, z2) ∧ F(x,y)) ∨ (¬ S(x, z1) ∨ ¬ S(y, z2)]

d) ∀x ∃y [(S(x, AUB) → x≠y ∧ S(y, AUB) ∧ ¬K(x, y)]
Every AUB student doesn't know at least one other AUB student

e) ∃z ∃x ∀y S(x,z) ∧ ¬F(x,y)
Some students in some universities are not Facebook friends with anyone

f) All Facebook friends aren't in any university
∀x∀y∀z1∀z2 F(x, y) → (¬ S(x, z1) ∧ ¬S(y,z2))

Exercise 11 (10 points)

Show that $\forall x P(x) \lor \forall x Q(x)$ and $\forall x \forall y (P(x) \lor Q(y))$, where all quantifiers have the same nonempty domain, are logically equivalent. (The new variable y is used to combine the quantifications correctly.)

To prove this true, we need to prove:

1. $\forall x P(x) \lor \forall x Q(x) \rightarrow \forall x \forall y (P(x) \lor Q(y))$
2. $\forall x \forall y (P(x) \lor Q(y)) \rightarrow \forall x P(x) \lor \forall x Q(x)$

1. $\forall x P(x) \lor \forall x Q(x)$ Premise
2. $P(c) \lor \forall x Q(x)$ U.I from 1
3. $P(c) \lor Q(d)$ U.I from 2
4. $\forall x P(x) \lor Q(d)$ U.G from 3
5. $\forall x \forall y (P(x) \lor Q(y)$ U.G from 4

Then $\forall x P(x) \lor \forall x Q(x) \rightarrow \forall x \forall y (P(x) \lor Q(y)$ is true

1. $\forall x \forall y (P(x) \lor Q(y))$ Premise
2. $\forall y (P(c) \lor Q(y))$ U.I from 1
3. $P(c) \lor Q(d)$ U.I from 2
4. $\forall x P(x) \lor Q(d)$ U.G from 3
5. $\forall x P(x) \lor \forall x Q(x)$ U.G from 4

Then $\forall x \forall y (P(x) \lor Q(y)) \rightarrow \forall x P(x) \lor \forall x Q(x)$ is true

Then they are logically equivalent

American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16
Assignment 3 Solution

Please solve the following exercises and submit **BEFORE 8:00 pm of Friday 9th, October.** Submit on Moodle.

Exercise 1 (10 points)

What is wrong with this argument? Let $S(x, y)$ be "x is smarter than y". Given the premise $\exists s\ S(s,\text{Rayyan})$, it follows that $S(\text{Rayyan, Rayyan})$. Then by existential generalization it follows that $\exists x\ S(x,x)$, so that someone is smarter than himself.

The wrong step is the conclusion $S(\text{Rayyan, Rayyan})$ from the given premise $\exists s\ S(s,\text{Rayyan})$…if there exist some s that satisfy $S(s,\text{Rayyan})$, this doesn't mean that any value for s work…So we can't replace it with Rayyan.

Exercise 2 (10 points)

Use inference rules to show that the following compound propositions are not satifiable

a) $(p \rightarrow q) \wedge (p \vee \neg q) \wedge (p \vee r) \wedge (\neg r \vee q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee \neg q)$

$\mathbf{(p \rightarrow q)} \wedge (p \vee \neg q) \wedge (p \vee r) \wedge (\neg r \vee q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee \neg q)$

$\leftrightarrow (\neg p \vee q) \wedge (p \vee \neg q) \wedge \mathbf{(p \vee r) \wedge (\neg r \vee q)} \wedge (\neg p \vee \neg r) \wedge (\neg p \vee \neg q)$

$\leftrightarrow (\neg p \vee q) \wedge \mathbf{(p \vee \neg q) \wedge (p \vee q)} \wedge (\neg p \vee \neg r) \wedge (\neg p \vee \neg q)$ – resolution

$\leftrightarrow \mathbf{(\neg p \vee q)} \wedge p \wedge (\neg p \vee \neg r) \wedge \mathbf{(\neg p \vee \neg q)}$ – resolution

$\leftrightarrow \mathbf{\neg p \wedge p} \wedge (\neg p \vee \neg r)$ – resolution

$\leftrightarrow \mathbf{F} \wedge (\neg p \vee \neg r)$ – contradiction

$\rightarrow$ Not Satisfiable

b) $(p \rightarrow q) \wedge (p \vee q) \wedge (\neg q) \wedge (p \vee \neg q)$

$\mathbf{(p \rightarrow q)} \wedge (p \vee q) \wedge (\neg q) \wedge (p \vee \neg q)$

$\leftrightarrow (\neg p \vee q) \wedge (p \vee q) \wedge (\neg q) \wedge (p \vee \neg q)$

$\leftrightarrow \mathbf{(\neg p \vee q) \wedge (p \vee q)} \wedge (\neg q) \wedge (p \vee \neg q)$

$\leftrightarrow \mathbf{q \wedge (\neg q)} \wedge (p \vee \neg q)$ – resolution

$\leftrightarrow \mathbf{F} \wedge (p \vee \neg q)$ – contradiction

$\rightarrow$ Not Satisfiable

c) $\mathbf{(p \rightarrow q)} \wedge \mathbf{(q \rightarrow r)} \wedge (r \vee p) \wedge (\neg q \vee \mathbf{p \rightarrow \neg r}) \wedge (\neg r \vee p) \wedge (\neg r \vee q)$

$\mathbf{(p \rightarrow q)} \wedge \mathbf{(q \rightarrow r)} \wedge (r \vee p) \wedge (\neg q \vee \mathbf{p \rightarrow r}) \wedge (\neg r \vee p) \wedge (\neg r \vee q)$

$\leftrightarrow \mathbf{(\neg p \vee q) \wedge (\neg q \vee r)} \wedge (r \vee p) \wedge (\neg q \vee \neg p \vee \neg r) \wedge (\neg r \vee p) \wedge (\neg r \vee q)$

$\leftrightarrow$ **$(\neg p \lor r) \land (r \lor p)$** $\land (\neg q \lor \neg p \lor \neg r) \land (\neg r \lor p) \land (\neg r \lor q)$ - resolution
$\leftrightarrow$ **$r \land (\neg r \lor p)$** $\land (\neg q \lor \neg p$ or $\neg r) \land (\neg r \lor q)$
$\leftrightarrow$ **$[(r \land \neg r) \lor (r \land p)]$** $\land (\neg q \lor \neg p \lor \neg r) \land (\neg r \lor q)$ – distribution
$\leftrightarrow$ **$[f \lor (r \land p)]$** $\land (\neg q \lor \neg p \lor \neg r) \land (\neg r \lor q)$
$\leftrightarrow$ $r \land p \land$ **$(\neg q \lor \neg p \lor \neg r) \land (\neg r \lor q)$**
$\leftrightarrow$ $r \land$ **$p \land (\neg p \lor \neg r)$** -- resolution
$\leftrightarrow$ $r \land$ [**$(p \land \neg p)$** $\lor (p \land \neg r)]$ -- distribution
$\leftrightarrow$ $r \land [F \lor (p \land \neg r)]$
$\leftrightarrow$ $r \land \neg r \land p$ -- Contradiction
$\rightarrow$ Not Satisfiable

Exercise 3 (10 points)

Determine whether these are valid arguments

a) Some even numbers are prime. 8 is an even number. Then **8 is prime**
Not valid. Can't conclude from for *there exists* to a value of our choice

b) The square of all real numbers is positive. $a^2 < 0$, then **a is not real**
Valid. $\forall x\ (R(x) \rightarrow x^2 \geq 0) \rightarrow \forall x\ (x^2 < 0 \rightarrow \neg R(x))$, where R(x) is "x is a real number"

c) $(p \lor q) \rightarrow (r \land s)$, $(p \land q) \rightarrow t$, $\neg t$ and conclusion **$\neg r \lor \neg s$**
$(p \land q) \rightarrow t$ & $\neg t$ then, $\neg(p \land q)$ Modes Tollens
$\neg(p \land q)$ nothing to conclude… $\rightarrow$ not valid (if p or q is true, the conclusion would be false)

d) "*If you are starving or sick you will not be able to focus*", "*Sleeping well is necessary and sufficient to be able to focus*", "*You can't sleep well but you aren't starving*" therefore "**you are sick**".
Statements:
1. starve $\lor$ sick $\rightarrow \neg$focus
2. sleep $\leftrightarrow$ focus
3. $\neg$sleep $\land \neg$starve

Conclusions:
1. $\neg$sleep –Simplification from 3
2. $\neg$focus – Biconditional in 2

Nothing more..so not valid

Exercise 4 (10 points)

American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16
Assignment 3 Solution

Use rules of inference to determine whether the following conclusions are valid or not:

a) Let P(x) be "*x is a photographer*," O(x) be "*x owns a camera*," J(x) be "*x is a judge in the photoshoot competition*"", and the premises: $\exists x(P(x) \wedge \neg O(x))$, $\forall x(P(x) \rightarrow J(x))$ with the conclusion $\mathbf{\exists x(J(x) \wedge \neg O(x))}$.

1. $\exists x(P(x) \wedge \neg O(x))$ — Premise
2. $P(a) \wedge \neg O(a)$ — Existential instantiation from (1)
3. $P(a)$ — Simplification from (2)
4. $\forall x(P(x) \rightarrow J(x))$ — Premise
5. $P(a) \rightarrow J(a)$ — Universal instantiation from (4)
6. $J(a)$ — Modus ponens from (3) and (5)
7. $\neg O(a)$ — Simplification from (2)
8. $J(a) \wedge \neg O(a)$ — Conjunction from (6) and (7)
9. $\exists x(J(x) \wedge \neg O(x))$ — Existential generalization from (8)

Valid

b) $\forall x\,(P(x) \vee Q(x))$, and the conclusion $\mathbf{\forall x\, P(x) \vee \forall x\, Q(x)}$

1. $\forall x\,(P(x) \vee Q(x))$ — Premise
2. $P(a) \vee Q(a)$ — Universal Instantiation
3. ?

It is not valid. Consider P(x): $x > 10$ and Q(x) = $x \leq 10$, and domain of x be all integers. $\forall x\,(P(x) \vee Q(x))$ is true, but $\forall x\, P(x) \vee \forall x\, Q(x)$ is false

c) $\forall x\, P(x) \vee \forall x\, Q(x)$, and the conclusion $\mathbf{\forall x\,(P(x) \vee Q(x))}$

1. $\forall x\, P(x) \vee \forall x\, Q(x)$ — Premise
2. $P(a) \vee Q(a)$ — Universal Instantiation
3. $\forall x\,(P(x) \vee Q(x))$ — Universal Generalization

Valid

d) $\forall x(P(x) \rightarrow (Q(x) \wedge S(x)))$ and $\forall x(P(x) \wedge R(x))$, and the conclusion $\mathbf{\forall x(R(x) \wedge S(x))}$.

Valid

e) "*Amine is a bad boy or Souad is a good girl*" and "*Amine is a good boy or Sarah is happy*" imply the conclusion "*Souad is a good girl and Sarah is happy*."

Valid

Exercise 5 (10 points)

For each of these arguments, explain which rules of inference are used for each step.

a) "*All self-centered people suffer from illusion of control*". "*Natasha, our classmate is self-centered*". Therefore, "**some of our classmates suffer from the illusion of control**".
S(x) = "x is self-centered", I(x) = "x suffers from illusion of control", C(x) = "x is in our class"
1. ∀x S(x) → I(x) Premise
2. C(Natasha) ∧ S(Natsha) Premise
3. S(Natasha) Simplification from 2
4. I(Natasha) Modus Ponens from 1 and 3
5. I(Natasha) ∧ S(Natsha) Conjugation from 3 and 4
6. ∃x I(x) ∧ S(x) E.G from 5

b) "*All foods that are healthy to eat do not taste good*." "*Tofu is healthy to eat*." "*You only eat what tastes good*." Therefore "**You do not eat tofu**."

c) "*If you are starving or sick you will not be able to focus*", "*Sleeping well is necessary and sufficient to be able to focus*", "*You aren't starving and you slept well*", therefore "**you aren't sick**".

Exercise 6 (10 points)

Prove or disprove each of the following conjunctures and mention what **type of proof** you used.

a) if a-b is odd, and b+c is odd, then **a+c is even**.
a-b is odd ➔ a-b = 2k +1
b+c is odd ➔ b+c = 2m+1
(a-b) + (b+c) = a+c = 2k+1 + 2m + 1 = 2(k+m+1), then it's even
Direct Proof

b) If a and b are 2 distinct prime numbers, then $\sqrt{\boldsymbol{a}\,\boldsymbol{b}}$ **is not an integer**
if a and b are 2 distinct prime numbers, then a ≠ b, and both a and b have no divisors except themselves and 1
Now assume that $\sqrt{\boldsymbol{a}\,\boldsymbol{b}}$ is an integer,
➔ $\sqrt{\boldsymbol{a}\,\boldsymbol{b}} = \boldsymbol{c},$ for some integer c,
➔ $(\sqrt{\boldsymbol{a}\,\boldsymbol{b}})^2 = c^2,$
➔ ab $= c^2,$
➔ $\frac{\text{ab}}{\text{c}} = c,$
but since c *(on the right)* is an integer, then c *(in the denominator)* must be able to divide a or b, then:

- c = a or
- c = b or
- a is not prime

- or b is not prime

If c = a or c = b, then a = b since ab $= c^2$, but a and b are distinct numbers
We are only left with a being not prime or b being not prime, which is a contradiction
Then $\sqrt{a\,b}$ **is not an integer**
Proof by contradiction and Proof by cases

c) *m*^2 = *n*^2 if and only if *m* = *n* or *m* = −*n*.
   1. *m*^2 = *n*^2 → *m* = *n* or *m* = −*n*
      m^2 = *n*^2 → |m| = |n|, then m = n or m = -n
      Direct Proof
   2. *m* = *n n* or *m* = −*n* → *m*^2 = *n*^2
      - m = n, then m^2 = n^2
      - m = -n then m^2 = (-n)^2 = n^2

      Proof by cases

d) *if a^2 is odd and b^2 is even, then* ***(a+b)^2 is odd***
   if a^2 is odd and b^2 is even,
   → a^2 + b^2 is odd, since odd + even is odd
   → a^2 + b^2 + 2ab is also odd since 2ab is even and (a^2 + b^2) is odd
   →(a+b)^2 is odd
   Direct Proof

e) *some integers are the sum of all integers preceding them*
   3 = 2 + 1 +0
   Direct Proof

Exercise 7 (10 points)

Prove the each of the following conjunctures by **contraposition and by contradiction**

a) if n^2 +1 is odd, then **n is even,** where n is an integer
   - **Contraposition:** *if n is odd, then n^2 + 1 is even*
     n = 2k+1, since n is odd
     → n^2 = 4k^2 + 4k + 1
     →n^2 + 1 = 4k^2 + 4k + 1 + 1 = 2(2k^2 + 2k + 1) which is even
   - **Contradiction:** assume n is odd and n^2 + 1 is also odd
     → n^2 is odd, then n^2 +1 is even; but we know that n^2 +1 is odd,
     → n^2 should be even and n is even

b) if 3n+2 is odd, then **n is odd**

c) if n is a perfect cube, then **n + 5 is not a perfect cube** *(by contradiction only is enough)*
   <u>RTP:</u> If n is perfect cube, then n+5 is not a perfect cube

Assume that their exists n such that it is a perfect cube, and n+5 is also a perfect cube

Then $n = a^3$, and $n+5 = b^3$, for some integers a and b

$b^3 - a^3 = (b - a)(b^2 + ba + a^2) = 5$

since a and b are integers, then (b-a) and $(b^2 + ba + a^2)$ are integers, and since 5 is prime, then on of those 2 cases must be correct:

- $(b - a) = 5$ and $(b^2 + ba + a^2) = 1$ ➔ b=5+a ➔ $0 = 24+15a+3a^2$, then a is not real ➔ contradiction
- $(b - a) = 1$ and $(b^2 + ba + a^2) = 5$ ➔ b=1+a ➔ $0= 2a^2 + 3a - 4$, then a = -2.3 or a = 0.8 ➔ a is not integer ➔ contradiction

Then if n is a perfect cube, n+5 can't be a perfect cube

Exercise 8 (10 points)

If you have a drawer that contains socks of 3 different colors (white, black, blue), how many socks should you draw to be sure to have a pair of socks of having same color? **Prove it**.

After picking 3 socks, the 3 socks might be 1 white, 1 black, and 1 blue; therefore no pair was drawn. Picking a fourth sock would for sure be either white, or black or blue, and therefore of those 3 colors will have 2 socks drawn from drawer and therefore a pair was surely found

Exercise 9 (10 points)

Prove that P(10) is true for P(n) = n^2 + 10 > 10n. What kind of proof did you use?

Direct proof

Prove that P(n) is true for all n >= 9. What kind of proof did you use?

Direct proof

Exercise 10 (10 points)

Show that the following statements are equivalent , give n is integer

1. $n^3 is\ even$
2. $1 - n\ is\ odd$
3. n/2 is an integer
4. $n^2 + 1\ is\ odd$

Need to show that 1↔2 ↔3 ↔4, or 1 ↔2 and 1 ↔ 3 and 1 ↔ 4

1 ↔ 2:

- n^3 is even → 1-n is odd
  By contraposition show that if 1-n is even, then n^3 is odd
  1-n is odd → 1-n = 2k, then n = -2k+1,
  then n^3 = (4k^2 -4k + 1)(-2k +1) = -8k^3 +8k^2 -2k + 4k^2 -4k +1
  Then n^3 = 2(-4k^3 + 6k^2 -3k) + 1, then its odd
  hence we conclude than if n^3 is even, then 1-n is odd
- 1-n is odd → n^3 is even
  1-n is odd, then n is even, then n^3 is even *(with a little more spices)*

1 ↔ 3:

- n^3 is even → n/2 is an integer
  we know that when n^3 is even, then n is even, then n = 2k, then n/2 = k which is an integer
- n/2 is an integer → n^3 even
  Similar proof

1 ↔ 4:

- n^3 is even → n^2 + 1 is odd
  Similar Proof
- n^2 + 1 is odd → n^3 even
  Similar proof

Exercise 11 (10 points)

Use a proof by cases to show that

a) max(*a*, max*(b, c))* = max(max*(a, b), c)* whenever *a*, *b*, and *c* are real numbers
   3 cases: a is the max, or b is the max, or c is the max
   Test the 2 functions for each case and they would produce same result, and that how you prove it

b) $|x| + |y| \geq |x + y|$ (where $|x|$ represents the absolute value of x, which equals x if $x \geq 0$ and equals $-x$ if $x < 0$) – Triangle Inequality.
   4 cases:
   1. x ≤ 0 and y ≤ 0
      $|x| + |y|$ = -x-y, and $|x + y|$ = -x-y, then $|x| + |y| \geq |x + y|$
   2. x > 0 and y > 0
      similar proof
   3. x ≤ 0 and y > 0
      then –x > 0
      $|x| + |y|$ = -x + y
      If $|x| \leq$ y, then $|x + y|$ = x + y < y < -x+y
      If $|x| \geq$ y, then $|x + y|$ = -x - y < y < -x + y

then $|x| + |y| \geq |x + y|$

4. $x > 0$ and $y \leq 0$

similar proof

Exercise 12 (10 points)

Formulate and proof a conjecture about the last digit (right most) of the 4th power of any integer.

$0^4 = 0$
$1^4 = 1$
$2^4 = 16$
$3^4 = 81$
$4^4 = 256$
$5^4 = 525$
$6^4 = 1269$
$7^4 = 2401$
$8^4 = 4096$
$9^4 = 6561$

Our conjecture is that any number raised to the fourth power will end with {0, 1,5,6, or 9}

We know that any integer can be written in the format 10x+i, where x is an integer, and i is an integer in the range [0-9]
Then for any n = 10x+i, n^4 = (10x + i)^4

$[(a + b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4 . \ . \ . \ . \ . \ ]$

Then n^4 = *something something* + i^4, and all other factors are multiplied by at least 10, and up to 10^4, so x would have no effect on the last digits for sure because of the multiplication of 10. Now since we just shown that all possible values of i^4 will end with {0, 1,5,6, or 9}, then our conjecture is proven true.

Assignment 4 Solution

Exercise 1:

Let p be you have hyperglycemia
Let q be you have hypertension
Let r be you have headaches
Let s be you have polydipsia
Let t be you feel dizzy

1.	$p \lor q \rightarrow r \land s$	Premise
2.	$p \lor q \rightarrow r$	Simplification (1)
3.	$r \rightarrow t$	Premise
4.	$\neg t$	Premise
5.	$\neg r$	Modus Tollens (3) and (4)
6.	$\neg(p \lor q)$	Modus Tollens (2) and (5)
7.	$\neg p \land \neg q$	De Morgan's Law (6)
8.	$\neg p$	Simplification (7)

Exercise 2:

a)

1.	Parallelogram → 2 pairs of parallel sides	Premise
2.	Square → parallelogram	Premise
3.	Trapezoid → 1 pair of parallel sides	Premise
4.	Quadrilaterals → 4 sides	Premise
5.	**¬(1 pair of parallel sides) → ¬Trapezoid**	**Contrapositive**
6.	**¬(2 pair of parallel sides) → ¬Parallelogram**	**Contrapositive**
7.	**Trapezoid → ¬Parallelogram**	**Hypothetical Syllogism (3) and (6)**
8.	**Parallelogram → ¬ Trapezoid**	**Hypothetical Syllogism (1) and (5)**
9.	**Square → 2 pairs of parallel sides**	**Hypothetical Syllogism (1) and (2)**

Conclusions: squares have 2 pairs of parallel sides. Trapezoids are not parallelograms. Parallelograms are not trapezoids. If we assume that parallelograms, squares and trapezoids are quadrilaterals, then by Hypothetical Syllogism also we can conclude that each of them have 4 sides.

b) Let W(x) be I work day x

Let S(x) be day x is sunny

Let P(x) be day x is partly sunny

1. ∀x (W(x) → S(x) ∨ P(x)) Premise
2. W(Monday) ∨ W(Friday) Premise
3. ¬S(Tuesday) Premise
4. ¬P(Friday) Premise
5. **S(Mon) ∨ P(Mon)) ∨ S(Fri) Modus Ponens (1) and (2)**
6. **(W(Tues) ∧ P(Tues)) ∨ (¬W(Tues)**

c) Let p be I am cheered up

Let q be I am upset

Let r be I make all people around me motivated

Let s be I make all people around me overjoyed

1. p ∨ q Premises
2. ¬ q Premises
3. **p Disjunction Syllogism (1) and (2)**
4. p → r ∨ s Premises
5. **r ∨ s Modus Ponens (3) and (4)**

Conclusions: I make all people around me motivated, overjoyed or both

d)

1. ∀x (Healthy(x) → ¬TastesGood(x)) Premise
2. Healthy(Tofu) → ¬TastesGood(Tofu) Universal Instantiation (1)
3. Healthy(Tofu) Premise
4. ¬TastesGood(Tofu) Modus Ponens (2) and (3)
5. ∀x (Eat(x) → TastesGood(x)) Premise
6. ¬Eat(Tofu) Premise
7. ¬Healthy(Cheeseburgers) Premise

Conclusions: Tofu does not taste good is concluded from (2) and (3). We can't conclude anything about cheeseburgers since we don't know if it tastes good or not, all what we know is that it is not healthy.

Exercise 3:

a) let R(x) be x is a real number
   1. $\forall x\ ((x^2 \neq 1) \land R(x) \rightarrow x \neq 1)$ Premise
   2. $R(a) \land (a^2 \neq 1) \rightarrow a = 1$? Premise
   3. $\exists x\ ((x^2 \neq 1) \land R(x) \rightarrow x = 1)$ Contradiction (1) and (3)

   ⇨ The argument is not valid

b) Let s be you are sick
   Let h be you are hungry
   Let c be you are feeling cold
   1. $s \land h \rightarrow c$ Premise
   2. $c \land h \rightarrow s$? Premise

   It is a fallacy to assume if c and h are true, we can conclude anything about s, since the c being true doesn't conclude anything…and h doesn't have any shown effect on s…

   ⇨ The argument is not valid

Exercise 4:

Line 3 and 5 have the same error (we can't use simplification since we have disjunction and not conjunction)

Exercise 5:

1. $\forall x\ (A(x) \lor B(x))$ Premise
2. $\forall x\ ((\neg A(x) \land B(x)) \rightarrow C(x))$ Premise
3. $\neg A(c) \land B(c)) \rightarrow C(c)$ Universal Instantiation (2)
4. $A(c) \lor \neg B(c) \lor C(c)$ Law of Implication (2)
5. $A(c) \lor B(c)$ Universal Instantiation (1)
6. $A(c) \lor C(c)$ Resolution (4) and (5)
7. $\neg C(c)$ Premise
8. $\neg C(c) \rightarrow A(c)$ Law of Implication (6)

9. $A(c)$ — Modus Ponens (7) and (8)
10. $\forall x\,(\neg C(x) \rightarrow A(x))$ — Universal Generalization (8)

Exercise 6:

1. $q \rightarrow (u \wedge t)$	Premise
2. $u \rightarrow p$	Premise
3. $q \rightarrow u \wedge t \wedge p$	Hypothetical Syllogism (1) and (2)
4. $(p \wedge t) \rightarrow (r \vee S)$	Premise
5. $q \rightarrow (r \vee S)$	Hypothetical Syllogism (3) and (4)
6. $\neg S$	Premise
7. $q \rightarrow r$	Disjunctive Syllogism (5) and (6)

Exercise 7:

a)

Samir is a student in this class.	Premises
Samir is from Russia.	Premises
Everyone from Russia has had a flu at least once.	Premises
Someone in this class has had a flu.	Existential Generalization

Let $S(x)$ be x is a student of this class
Let $R(x)$ be x is from Russia
Let $F(x)$ be x has had a flu

1. $S(Samir)$	Premise
2. $R(Samir)$	Premise
3. $\forall x\,(R(x) \rightarrow F(x))$	Premise
4. $R(Samir) \rightarrow F(Samir)$	Universal Instantiation (3)
5. $F(Samir)$	Modus Ponens (2) and (4)
6. $S(Samir) \wedge F(Samir)$	Conjunction (1) and (5)
7. $\exists x\,(S(x) \wedge F(x))$	Existential Generalization (6)

b) Each of five roommates, Melissa, Aaron, Ralph, Veneesha, and Keeshawn, has taken CMPS 200.
Every student who has taken CMPS 200 can take CMPS 212.

All five roommates can take CMPS 212

Let CMPS200(x) be x has taken CMPS 200
Let CMPS212(x) be x can take CMPS 212

1. CMPS200(Melissa) ∧ CMPS200(Aaron) ∧ CMPS200(Ralph) ∧ CMPS200(Veneesha) ∧ CMPS200(Keeshawn) — Premises
2. ∀x (CMPS200(x) → CMPS212(x)) — Premises
3. CMPS200(Melissa) → CMPS212(Melissa) — Universal Instantiation (2)
4. CMPS200(Aaron) → CMPS212(Aaron) — Universal Instantiation (2)
5. CMPS200(Ralph) → CMPS212(Ralph) — Universal Instantiation (2)
6. CMPS200(Veneesha) → CMPS212(Veneesha) — Universal Instantiation (2)
7. CMPS200(Keeshawn) → CMPS212(Keeshawn) — Universal Instantiation (2)
8. CMPS200(Melissa) — Simplification (1)
9. CMPS200(Aaron) — Simplification (1)
10. CMPS200(Ralph) — Simplification (1)
11. CMPS200(Veneesha) — Simplification (1)
12. CMPS200(Keeshawn) — Simplification (1)
13. CMPS212(Melissa) — Modus Ponens (3) and (8)
14. CMPS212(Aaron) — Modus Ponens (4) and (9)
15. CMPS212(Ralph) — Modus Ponens (5) and (10)
16. CMPS212(Veneesha) — Modus Ponens (6) and (11)
17. CMPS212(Keeshawn) — Modus Ponens (7) and (12)
18. CMPS212(Melissa) ∧ CMPS212(Aaron) ∧ CMPS212(Ralph) ∧ CMPS212(Veneesha) ∧ CMPS212(Keeshawn) — Conjunction (13), (14), (15), (16), (17)

c) All self-centered people suffer from the illusion of control. — Premises
Natasha, our classmate is self-centered. — Premises
Some of our classmates suffer from the illusion of control. — Existential Generalization

Let S(x) be x is self-centered
Let C(x) be x is in our class
Let I(x) be x suffers from the illusion of control

1. ∀x (S(x) → I(x)) — Premises

2. S(Natasha) → I(Natasha) Universal Instantiation (1)
3. S(Natasha) Premises
4. C(Natasha) Premises
5. I(Natasha) Modus Ponens (2) and (3)
6. C(Natasha) ∧ I(Natasha) Conjunction (4) and (5)
7. ∃x (C(x) ∧ I(x)) Existential Generalization (6)

d) All Math classes taught by professor John are wonderful. Premises
Professor John is giving a course this semester. Premises
The Math department has wonderful courses this semester. Modus Ponens

Let J(x) be John is giving course x
Let W(x) be course x is wonderful
The domain of x is all math courses

1. ∀x (J(x) → W(x)) Premises
2. ∃x (J(x)) Premises
3. J(c) → W(c) Universal Instantiation (1)
4. J(c) Universal Instantiation (2)
5. W(c) Modus Ponens (3) and (4)
6. ∃x (W(x)) Existential Generalization (5)

<u>Exercise 8:</u>

The problem in concluding B(George, George) given "there exists an s such that B(s, George)", is in the *there exists* itself, since this means that its is true from some *s,* and not for any value of *s*, so replacing *s* by George is fallacious.

<u>Exercise 9:</u>

a) Correct
1. ∀x (EnrolledInUni(x) → Dorm(x)) Premises
2. EnrolledInUni(Mia) → Dorm(Mia) Universal Instantiation (1)
3. ¬Dorm(Mia) Premises
4. **¬EnrolledInUni(Mia) Modus Tollens**

b) Not Correct
1. TouchScreen → Fragile Premises

2. ¬TouchScreen — Premises

We can't conclude anything about the laptop since the hypothesis is false (the conclusion can have any truth value)

c) Not Correct

1. Android → Java — Premises
2. Java — Premises

We can't conclude anything about the phone since the conclusion is true (the hypothesis can have any truth value)

d) Correct

1. Plant → ProduceGlucose — Premises
2. Palm → Plants — Premises
3. **Palm → ProduceGlucose** — **Hypothetical Syllogism**

Exercise 10:

1. (p ∨ q) ∧ (¬p ∨ q) ∧ (p ∨ ¬q) ∧ (¬p ∨ ¬q)	Premises
2. (p ∨ q)	Simplification (1)
3. (¬p ∨ q)	Simplification (1)
4. (p ∨ ¬q)	Simplification (1)
5. (¬p ∨ ¬q)	Simplification (1)
6. q	Resolution (2) and (3)
7. ¬q	Resolution (4) and (5)
8. q ∧ ¬q	Conjunction (6) and (7)
9. False (not satisfiable)	Negation Law (8)

American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16
Assignment 5

Please solve the following exercises and submit **BEFORE 11:55 pm of Monday 2nd of November.**

Please get a hardcopy submission whether you plan to solve it on a computer or on a paper. In Both cases, also submit to Moodle. However, if you submit a handwritten solution, I will only correct the questions that I manage to read (and easily find).

You can submit the hard copies during by Wednesday during the recitation. I will use Moodle submissions as a proof of early submissions. Don't try to modify anything in the hard copy submission, or else it will be considered cheating and you'll get a Zero.

Exercise 1 (10 points)

Can you guess the next number?

2 6 18 54 162 ...

a) Show that the value of the nth element $a_n = 3^n - 3^{n-1}$

$$a_n = 3^{n-1}(3-1) = 3^{n-1} * 2$$
$$a_1 = 3^{1-1} * 2 = 1 * 2 = 2$$
$$a_2 = 3^{2-1} * 2 = 3 * 2 = 6$$
$$a_3 = 3^{3-1} * 2 = 9 * 2 = 18$$

b) Find an equation for $S_n = a_1 + a_2 + a_3 + .. + a_i + \cdots + a_n$

$$S_n = a_1 + a_2 + a_3 + .. + a_i + \cdots + a_n$$
$$S_n = 3^n - 1$$

c) Prove the formula you conjectured in part (b).

Basic Step:

$$S_1 = 3^1 - 1 = 2, \text{Valid}$$
$$S_2 = 3^2 - 1 = 8, \text{Valid}$$

Inductive Step:

$$S_n = 3^n - 1, \text{then } S_{n+1} = 3^{n+1} - 1$$
$$S_{n+1} = a_{n+1} + S_n$$

$S_{n+1} = 3^{n+1} - 3^n + 3^n - 1 = 3^{n+1} - 1$ Proved

Exercise 2 (10 points)

Prove that 6 divides 3^n - 3 whenever n is an integer > 0.

Basic Step:

Consider $f(n) = 3^n - 3$,
f(1) = 0, which is divisible by 6
f(2) = 9 – 3 = 6, which is divisible by 6
f(3) = 27 – 3 = 24, which is divisible by 6

Inductive step:

$f(n) = 3^n - 3$ is divisible by 6, then f(n+1) is divisible by 6
$f(n+1) = 3^{n+1} - 3$
$= 3*(3^n) - 3$
$= 3*3^n - 9 + 6$
$= 3*(3^n - 3) + 6$
$= 3*f(n) + 6$
= 3 * 6k + 6, since f(n) is divisible by 6
= 6 (3k + 1), which is divisible by 6
Proved by Induction!

It can be also proved without induction by saying that 3^n-3 is $3(3^{n-1} - 1)$, and $3^{n-1} - 1$ is always divisible by 2 since 3^x is always odd for any positive integer x, and thus 3^n-3 is divisible by 3 and 2 and then it is divisible by 6

Exercise 3 (10 points)

What is wrong with this "proof"?

- "Theorem" For every positive integer n, $\sum_{i=1}^{n} i = \frac{\left(n+\frac{1}{2}\right)^2}{2}$.
- Then $\sum_{i=1}^{k+1} i = \left(\sum_{i=1}^{k} i\right) + (k+1)$.

- By this inductive hypothesis, $\sum_{i=1}^{k+1} i = \frac{\left(k+\frac{1}{2}\right)^2}{2} + k + 1 = \frac{\left(k^2+k+\frac{1}{4}\right)}{2} + k + 1 = \frac{\left(k^2+3k+\frac{9}{4}\right)}{2} = \frac{\left(k+\frac{3}{2}\right)^2}{2} = \frac{\left[(k+1)+\frac{1}{2}\right]^2}{2}$, completing the inductive step.

For $n = 1$, $\sum_{i=1}^{n} i = 1$, and $\frac{\left(1+\frac{1}{2}\right)^2}{2} = \frac{2.25}{2}$, and thus the basic step doesn't hold

Exercise 4 (10 points)

Suppose that m and n are positive integers with m > n and f is a function from {1,2,...,m} to {1,2,...,n}. Use mathematical induction on the variable n to show that f is not one-to-one. [***Hint:*** *apply induction on n]*

Basic Step:
For $n = 1$ and any value of $m > n$, then f maps from {1, 2, ..m} to {1}, then multiple values in domain maps to {1}, and thus f is not one-to-one

Inductive step:
For any arbitrary n and m, such that $m > n$, f is not one-to-one
For n+1, such that $n+1 < m$, f maps from {1, 2, ..., m} to {1, 2, ..., n, n+1}:

- If no value in domain maps to n+1, then f maps {1, 2, ..., m} to {1, 2, ..., n} is not one to one by inductive hypothesis.
- If some value in {1, 2, ..., m} maps to n+1, call it *i*, then *i* maps to *n+1*, then we can remove "swap" *i* by *m* and remove *m* from the domain; so {1, 2, ... m-1} maps to {1, 2, ...n}, and $m-1 > n$ since $(n+1 < m)$, which is not one-to-one also by inductive hypothesis

Then f is not one-to-one

Exercise 5 (10 points)

In computer science, a binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child (https://en.wikipedia.org/wiki/Binary_tree).

A Ternary tree is similar to a binary tree, however instead of 2 children, each node can have up to 3 children.

American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16
Assignment 5

A prefect Tree is a tree such that all leaf nodes are of same depth, and all other nodes are full nodes; i.e: each node in a perfect ternary tree of depth h has 3 children, except for the nodes at depth h (leaf nodes) who have 0 children.

a) Formulate a conjecture about the number of nodes in a Perfect Ternary tree. You may assume that the smallest perfect Ternary tree has 1 single node, and height 0

For h = 0, total number of nodes in a ternary tree is 1
For h = 1, total number of nodes in a ternary tree is 4
For h = 2, total number of nodes in a ternary tree is 13
We can say that for an arbitrary height h, total number of nodes is

$$\text{nodes(h)} = \sum_{i=0}^{h} 3^{i} = \frac{1-3^{h+1}}{1-3} = \frac{3^{h+1}-1}{2}$$

b) Prove it using induction.

Inductive step:
Any perfect ternary tree of height h can be replicated 3 times, and joined by a common root, to create a perfect ternary tree (since the 3 subtrees of the root are subtrees) of height h+1

Based on inductive hypothesis, $\text{nodes}(h+1) = \frac{3^{h+2}-1}{2}$

Since each subtree is of height h, then the total number of nodes in each is nodes(h), thus the total number of nodes in a tree of height h+1 is:
$\text{nodes}(h+1) = 3 * \text{nodes}(h) + 1$, (1 is the root node)

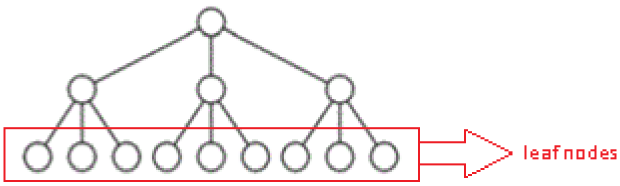
$$\text{nodes}(h+1) = 3 * \frac{3^{h+1}-1}{2} + 1$$

$$= \frac{3^{h+2}-3}{2} + 1$$

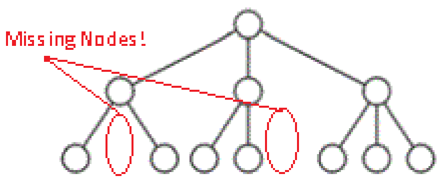
$$= \frac{3^{h+2}-1-2}{2} + 1$$

$$= \frac{3^{h+2}-1}{2} + 1 - 1 = \frac{3^{h+2}-1}{2}$$

Then the total number of nodes in a perfect ternary tree is $\frac{3^{h+1}-1}{2}$. Proved by Induction



A Perfect Ternary Tree of Height 3



A non-perfect Ternary Tree

Exercise 6 (10 points)

Assume that a chocolate bar consists of n squares arranged in a rectangular pattern. The entire bar can be broken along a vertical or a horizontal line separating the squares to get 2 rectangular pieces. Assuming that only one piece can be broken at a time:

a) Determine how many breaks you must successively make to break the bar into n separate squares
   if n = 1, you need 0 breaks
   if n = 2, you need 1 break
   if n = 3, you need 2 breaks
   if n = 4, you need 3 breaks.
   So for any bar of n pieces, we need n-1 breaks.
   B(n) = n-1
b) Use strong induction to prove your answer
   Basic step:

Shown in part a.
for all j / 0 < j <= n, B(n) = n-1

Inductive step:

Given a bar of n+1 pieces *[we expect B(n+1) = n]*, cutting it once forms 2 bars of size a and b, such that a+b = n+1, a & b >= 1, and a & b are integers; and thus 0 < a <= n, and 0 < b <= n.

Thus by inductive hypothesis, B(a) = a-1, and B(b) = b-1

Now B(n) = B(a) + B(b) + 1 (+1 is the first cut)

B(n+1) = a-1 + b-1 + 1 = a + b -1 = n;

Exercise 7 (10 points)

Consider the proposition $2^n > n^3$

a) Find an integer N such that the proposition is true whenever n is greater than N.

For n = 1, $2^1 > 1^3$, is false
For n = 2, $2^2 > 2^3$, is false
For n = 3, $2^3 > 3^3$, is false
For n = 4, $2^4 > 4^3$, is false
For n = 5, $2^5 > 5^3$, is false
For n = 6, $2^6 > 6^3$, is false
For n = 7, $2^7 > 7^3$, is false
For n = 8, $2^8 > 8^3$, is false
For n = 9, $2^9 > 9^3$, is false
For n = 10, $2^{10} > 10^3$, is 1024 > 1000, which is true
And thus for all n >= 10, the proposition is true

b) Prove that your result for all n > N using mathematical induction.

Basic step:

Shown in part a.

$P(n) = 2^n > n^3$, is true for all n >= 10

Inductive Step:

$$P(n+1) = 2^{n+1} > (n+1)^3\ ?$$

$$2^{n+1} = 2 * 2^n > 2n^3, \text{ using inductive hypothesis}$$

$$\leftrightarrow 2^{n+1} > 2n^3$$

$$\leftrightarrow 2^{n+1} > n^3 + n^3$$

$$\leftrightarrow 2^{n+1} > n^3 + 6n^2, \text{ since n >=10}$$

$$\leftrightarrow 2^{n+1} > n^3 + 3n^2 + 10n, \text{ since n >=10}$$

$$\leftrightarrow 2^{n+1} > n^3 + 3n^2 + 3n + 7n$$

$$\leftrightarrow 2^{n+1} > n^3 + 3n^2 + 3n + 1$$

$$\leftrightarrow 2^{n+1} > (n+1)^3$$

OR

$$2^{n+1} = 2 * 2^n > 2n^3$$

$$\text{Is } 2n^3 > (n+1)^3?$$

Basic steps for n >=10 works

Inductive step:

$$2(n+1)^3 > (n+2)^3?$$

$$2(n+1)^3 = 2n^3 + 6n^2 + 6n + 2$$

$$(n+2)^3 = n^3 + 6n^2 + 12n + 8$$

$$2(n+1)^3 - (n+2)^3 = n^3 - 6n - 6 = n(n^2 - 6) - 6 >?\ 0$$

$$n \geq 10, \text{then } n^2 - 6 \geq 96$$

$$n(n^2 - 6) - 6 \geq 0$$

Then

$$2(n+1)^3 > (n+2)^3$$

$$\text{So } 2^{n+1} > 2n^3 > (n+1)^3$$

Exercise 8 (10 points)

Assume you can only use 5-cent and 9-cent stamps.

a) Determine which amounts of postage can be formed by the given stamps
We need n = 5a + 9b such that a, b >= 0, and a and b are integers
Thus we can form postage of value 5, 9, 10, 14, 15, 18, 19, 20, 23, 24, 25, 27, 28, 29, 30**, 32, 33, 34, 35, 36, 37, 38, 39, 40…**

b) Prove your answer to (a) using the principle of mathematical induction. Be sure to state explicitly your inductive hypothesis in the inductive step.
Basic step:
Using stamps of 5 and 9 cents only, we were able to make postages with amounts = **32, 33, 34, 35, 36, 37, 38, 39, 40…** as shown in part a

Inductive step:
Using stamps of 5 and 9 cents, we can create postage of amount n >= 32
In other words, n = 5a + 9b for n >=32, a,b >= 0 and a, b & n are integers.
So by inductive hypothesis, there is a configuration of a and b such that the above conditions are true, then n = 5a + 9b, for some value of n >=32.

We need to show that we can form n+1, such that n+1 = 5a' + 9b', with conditions for a' and b' similar to those of a and b

$$n+1=\begin{cases}5(a+2)+9(b-1)\\5(a-7)+9\,(b+4)\end{cases}$$

then [a' = a+2 and b' = b-1], or [a' =a-7 and b'=b+4]
notice that 5a' + 9b' in both cases is equals to n+1

Now case 1: if n = 5a + 9b has b >=1, then b' = b-1 >=0, then b' is valid; hence we can form n+1 = 5(a+2) + 9(b-1); i.e: by adding 2 5-cent stamps, and removing 1 9-cent stamp.

The other case would be that n = 5a + 9b and b = 0, then n = 5a, we know that n>= 32, then 5a >= 32, then a >= 32/5 → a >= 6.4, but a is an integer, so a >= 7, then a-7 >= 0, then a'= a-7 is also valid; hence we can form n+1 = 5(a-7) + 9(b+4); i.e: by removing 7 5-cents stamps, and adding 6 9-cent stamps
Then we can get n+1 if we have n formed of 5-cents and 9-cents stamps and n >= 32
Proved by induction.

c) Prove your answer to (a) using strong induction. How does the inductive hypothesis in this proof differ from that in the inductive hypothesis for a proof using mathematical induction?
We know from basic step shown above that the there is a configuration of a and b for n = 32, 33, 34, 35, and 36.
So for all j/ 32<= j <= n, and 36 <= n the conjecture above holds.

To get n+1, we simply add a 5-cents stamp to the configuration of n-4, giving a total of n+1 cents. We know that n-4 has a valid configuration by inductive hypothesis since 36 <= n, then 32 <= n -4 <= n (j = n-4)
So, since n-4 can be made of 5-cents and 9-cents stamps, adding a 5-cent stamp will give a total of n+1 cents made of 5 and 9 cents stamps also.
Proved by Strong Induction.!

Exercise 9 (10 points)

Use strong induction to show that every positive integer n can be written as a sum of distinct powers of two, that is, as a sum of a subset of the integers $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, and so on.
*[**Hint:** For the inductive step, separately consider the case where k + 1 is even and where it is odd. When it is even, note that (k + 1)/2 is an integer.]*

Basic Step:
$1 = 2^0$, this is a sum of distinct powers of 2
$2 = 2^1$, this is a sum of distinct powers of 2
$3 = 2^0 + 2^1$, this is a sum of distinct powers of 2
$4 = 2^2$, this is a sum of distinct powers of 2

Inductive step:
for all j/ 1<= j <= k, for an arbitrary k, j can be written as a sum of distinct powers of 2
We need to show that this property holds for k+1
If k is even, then 2^0 isn't present in the sum since 2^0 is the only odd number in all powers of 2 and k is even, then we can form k+1 by adding 2^0 to the sum of distinct powers, keeping them distinct
The other case would be that k is odd, in which it would be the case that k+1 is even, and (k+1)/2 is an integer in the range 1<= (k+1)/2 <= k, then (k+1)/2 is a sum distinct powers of 2 by inductive hypothesis. Adding 1 to each power of 2 (i.e: left shifting the powers) will keep the powers of 2 distinct and will give us 2*(k+1)/2, which is k+1
Therefore, the predicate is proved by Strong induction

American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16
Assignment 6

Please solve the following exercises and submit **BEFORE 7:00 pm of Friday 6th of November.**

Please get a hardcopy submission whether you plan to solve it on a computer or on a paper. In Both cases, also submit to Moodle. However, if you submit a handwritten solution, I will only correct the questions that I manage to read (and easily find).

You can submit the hard copies during by Wednesday during the recitation. I will use Moodle submissions as a proof of early submissions. Don't try to modify anything in the hard copy submission, or else it will be considered cheating and you'll get a Zero.

Exercise 1 **(10 points)**

Sort the following running times in ascending order; if two running times have the same growth rate, put an equal sign between them instead of <

- Sqrt(n)
- $\log(n^2)^{100}$
- $n^{1.01}$
- $n^2/1000$
- 2^2
- 3^{n-1}
- 2n-5
- $n^2\log(n/4)$
- $(n^2+5)/(n-5)$
- $n\log(n^2)$
- $n^2/\log(n)$

$2^2 < \log(n^2)^{100} < Sqrt(n) < 2n-5 <= (n^2+5)/(n-5) < n\log(n^2) < n^{1.01} < n^2/\log(n) < n^2/1000 < n^2\log(n/4) < 3^{n-1}$

Exercise 2 **(10 points)**

Show how the following list of characters gets sorted **step by step** using each of the following sorting algorithms: {c, a, f, d, e}

a) Bubble Sort

First Loop:

Iteration 1.

Second loop:

1. {**c, a**, f, d, e}
2. {a, **c, f**, d, e}
3. {a, c, **f, d**, e}
4. {a, c, d, **f, e**}
5. {a, c, d, e, f}

Iteration 2. {a, c, d, e, f}
Iteration 3. {a, c, d, e, f}
Iteration 4. {a, c, d, e, f}
Iteration 5. {a, c, d, e, f}

Done

b) Insertion Sort

First Loop:

Iteration 1.

Second loop:

1. {*c*, **a**, f, d, e}
2. {*a, c*, f, d, e}

Iteration 2. {*a, c, **f***, d, e}
Iteration 3. {*a, c, **d**, f*, e}
Iteration 4. {*a, c, d, **e**, f*}

Done

Exercise 3 (15 points)

Determine whether each of these functions is $O(x^3)$, $\Omega(x^3)$ and $\theta(x^3)$.

a) $f(x) = 17x+11$

- $17x + 11 < 17x^3 + 11x^3 <= 28x^3$, for $x > 0$,

So f(x) is $O(x^3)$ for $k = 1, C = 28$

- Assume $Cx^3 <= 17x + 11$ for all $x >= k > 0$, for some value of k, $C > 0$

➔ $Cx^2 < 17 + 11/x$, $\lim_{x\to\infty} Cx^2 = \infty$, $\lim_{x\to\infty} 17 + 11/x = 17$,

➔ $\infty < 17$, which is false, so f(x) is not $\Omega(x^3)$

- f(x) is not $\theta(x^3)$

b) $f(x) = x^3+1000x$

- $x^3+1000x <= x^3 + 1000x^3 <= 1001x^3$, then f(x) is $O(x^3)$ for $C = 1001$, $k = 1$

- $x^3 <= x^3+1000x$, so f(x) is $\Omega(x^3)$ for C = 1, k = 1
- Since f(x) is both $O(x^3)$ and $\Omega(x^3)$, then f(x) is $\theta(x^3)$

c) $f(x)= x \log x + 100x$
for x > 2,
- $x \log x + 100x <= x^2 \log x + 100x^2 <= x^3 + 100x^3 <= 101x^3$, So f(x) is $O(x^3)$ for k = 2, C = 101
- Assume $Cx^3 <= x\log x + 100x$ → $Cx^2/\log x <= 1 + 100/\log x$. It is clear that asymptotically $Cx^2/\log x$ will give +infinity, while $1 + 100/\log x$ will give 1, so this can't be true. so f(x) is not $\Omega(x^3)$
- Then f(x) is not $\theta(x^3)$

d) $f(x)= x^4/2$
- Assume $x^4/2 <= Cx^3$, then $x<2C$, which can't be since the inequality should hold for all x > k, so f(x) is not $O(x^3)$
- $Cx^3 <= x^4/2$, for c = 0.5, $x^3 < x^4$ which is clearly true, so f(x) is $\Omega(x^3)$
- Then f(x) is not $\theta(x^3)$

e) $f(x)= 2^x$
- Assume $2^x < Cx^3$, then $\log(2^x) < \log(Cx^3)$, then $x < \log C + 3\log x$, then $x/3\log x < (\log C+1)/3\log x$… asymptotically is 0, however, $x/3\log x$ can be shown to be +infinity by applying L'Hopital's rule., so its false, so f(x) is not $O(x^3)$
- $Cx^3 < 2^x$ can be similarly be processed to actually show that it is true, for C = 0.25, x > 2 , so f(x) is $\Omega(x^3)$
- Then f(x) is not $\theta(x^3)$

f) $f(x)= \lfloor x \rfloor \cdot \lceil x \rceil \cdot x$
a <= x <= a+1, where a is an integer, then:
$a <= \lceil x \rceil <= a+1$ and
$a <= \lfloor x \rfloor <= a+1$
➔ $a^3 <= \lfloor x \rfloor \cdot \lceil x \rceil \cdot x <= (a+1)^3 = a^3 + 6a^2 + 6a + 1 <= 14a^3$
- So f(x) is $O(x^3)$ for C = 14, k = 1
- And f(x) is $\Omega(x^3)$ for C = 1, k = 1
- Since f(x) is both $O(x^3)$ and $\Omega(x^3)$, then f(x) is $\theta(x^3)$

Exercise 4 (10 points)

Give a big-O estimate for each of these functions. For the function *g* in your estimate O(g(x)), use a simple function *g* of smallest order.

a) $(n^3 + n^2 \log n)(\log n - 1) + (11 n \log n + 9)(n^3 + 2)$

$= n^3 \log n + ... + 11n^4 \log n$
$= O(n^4 \log n)$

b) $(3^n + n^2)(n^3 + 2^n)$
$= n^3 * 3^n + n^5 + 6^n + n^2 * 2^n$
$= O(6^n)$

c) $(n^n + n^{2n} + 5n)(n! + 5^n)$
$= n! * n^n + 5^n * n^n + n! * n^{2n} + 5^n * n^{2n} + n! * 5n + 5^n * 5n$
$= O(n! * n^{2n})$

Exercise 5 (20 points)

Given a list of unsorted numbers of n elements, devise ~~linear time~~ algorithms that:

a) locates the position of the element with value closest to the arithmetic mean of the list.

```
Closest_to_mean(a1, a2, a3, a4, ..., an):
    //calculate mean
    mean = 0
    for i = 1 → n
        mean = mean + ai
    mean = mean / n

    //find location of num closest to mean
    loc = 1
    for i = 2 → n
        if |ai – mean| <= |aloc – mean|
            loc = i
    return loc
```

b) determines the range (difference between min and max) of a list of numbers.

```
range(a1, a2, a3, a4, ..., an):
    //find min and max
    min = a1
    max = a1
    for i = 2 → n
        if ai < min
            min = ai

        if ai > max
```

```
            max = a_i

    //compute and return range
    return max - min
```

c) finds all terms of a finite sequence of integers that are greater than the product of all previous terms of the sequence.

```
great((a_1, a_2, a_3, a_4, ..., a_n):
    product = 1
    for i = 1 → n
        if product < a_i
            print a_i
        product = product * a_i
```

d) produces as output the largest sum obtained by adding an integer in the list to the one following it.

```
largestSum(a_1, a_2, a_3, a_4, ..., a_n):
    //find min and max
    If n = 1
        return a_1

    else
        max = a_1 + a_2
        for i = 2 → n
            if a_(i-1) + a_i > max
                max = a_(i-1) + a_i
    return max
```

~~Exercise 6 (10 points)~~

~~Write a logarithmic algorithm for calculating x^n using successive multiplication, where n is a perfect power of 3.~~

Exercise 7 (15 points)

Assume that you were in an exam which you know all of its answers but don't have time to solve it all. Assume also that each question q_i requires a duration of t_i seconds to solve and has a total of p_i points.

a) Devise a greedy algorithm to get the highest grade if given time T for the exam, and assuming that you will get partial credit for partial answers.

b) Will a greedy strategy work if partial credits weren't given to partial answers? (Answer either get full credit or no credit at all). Explain.

Exercise 8 (10 points)

The ternary search algorithm locates an element in a list of increasing integers by successively splitting the list into three sublists of equal (or as close to equal as possible) size, and restricting the search to the appropriate piece.

Specify the steps of this algorithm. ~~What is the running time of this algorithm?~~

```
⌊⌈x⌉⌋
procedure ternary search(s: integer, a1,a2, . . . , an: increasing integers)
    i := 1
    j := n
    while i < j − 1
        first := ⌊ (i + j)/3⌋
        second := ⌊2(i + j)/3⌋
        if x > au then
            i := second + 1
        else if x > a_first then
            i := first + 1
            j := second
        else
            j := first

    if x = ai then
        location := i
    else if x = aj then
        location := j
    else
        location := 0

    return location {0 if not found}
```

Exercise 9 (10 points)

Determine whether

a) n log n is O(log n!)

$nlgn \leq C * log(n!)$

$2^{nlgn} \leq 2^{c*logn!}$

$2^{nlgn} \leq 2^{c*logn!}$

$n^n \leq (n!)^c$~~, for c >= n, this is true~~

So n log n is O(log n!)

http://ballinger.disted.camosun.bc.ca/math126/lognfactorial.pdf

b) log(n + 1) is O(log n)
same idea, we'll get $n+1 \leq n^c$, which is true for c = 2, k = 2

c) $\log(n^2 + 1)$ is O(log n)
same idea, we'll get $n^2+1 \leq n^c$, which is true for c = 3, k = 1

~~**Exercise 10 (15 points)**~~

~~Give a big-*O* estimate for the number additions used in each of the following segments of algorithms.~~

~~a) *t* := 0~~
~~**for** *i* := 1 **to** *n*~~
~~**for** *j* := 1 **to** *n*~~
~~*t* := *t* + *i* + *j*~~

~~b) *m* := 0~~
~~**for** *i* := 1 **to** *n*~~
~~**for** *j* := *i* + 1 **to** *n*~~
~~*m* := max(*aiaj*,*m*)~~

~~(where *a*1, *a*2, ..., *an* are positive real numbers)~~

~~c) *i* := 1~~
~~*t* := 0~~
~~**while** *i* ≤ *n*~~
~~*t* := *t* + *i*~~
~~*i* := 2*i*~~

American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16
Assignment 7 Solution

Please solve the following exercises and submit **BEFORE 11:55 pm of Sunday 29th of November.**

Please get a hardcopy submission whether you plan to solve it on a computer or on a paper. In Both cases, also **submit to Moodle**. However, if you submit a handwritten solution, I will only correct the questions that I manage to read (and easily find).

Pass the hardcopy submissions during the week, starting from Monday 30th. I will use Moodle submissions as a proof of early submissions. Don't try to modify anything in the hard copy submission, or else it will be considered cheating and you'll get a Zero.

Exercise 1 (15 points)

Give a big-*O* estimate for the number additions used in each of the following segments of algorithms.

a) *t* := 0
for *i* := 1 **to** *n*
    for *j* := 1 **to** *n*
        t := *t* + *i* + *j*

t := 0
for *i* := 1 **to** *n* *//n iterations*
    for *j* := 1 **to** *n* *//n iterations*
        t := *t* + *i* + *j* *//single computation*
$O(n^2)$

b) *m* := 0
for *i* := 1 **to** *n*
    for *j* := *i* + 1 **to** *n*
        m := max(*aiaj,m)*

(where *a*1, *a*2, ..., *an* are positive real numbers)
m := 0
for *i* := 1 **to** *n* *//n iterations*
    for *j* := *i* + 1 **to** *n* *//n – i -1 iterations*
        m := max*(aiaj,m)*

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n} 1 \approx \frac{n*(n+1)}{2} = O(n^2)$$

c) $i := 1$
$t := 0$
while $i \leq n$
    $t := t + i$
    $i := 2i$

$i := 1$
$t := 0$
while $i \leq n$ *//logn iterations*
    $t := t + i$
    $i := 2i$

$$\sum_{\substack{i=1 \\ i=i*2}}^{i\leq n} 1 = logn$$

Exercise 2 **(10 points)**

Write a logarithmic iterative algorithm for calculating x^n using successive multiplication, where n is a perfect power of 3.

```
procedure power(x, n)
{
        pow = x
        i = 1
        while i < n
                pow = pow * pow * pow
                i = 3 * i

        return pow
}
```

Exercise 3 **(10 points)**

The ternary search algorithm locates an element in a list of increasing integers by successively splitting the list into three sublists of equal (or as close to equal as possible) size, and restricting the search to the appropriate piece.

Specify the steps of this algorithm. What is the running time of this algorithm?

```
int ternary_search(int A[],int key)
{
        n=A.length;
        left =1;
        right =n;
        int third1, third2=0;
        While i<j
        {
                third1= ⌊ (left + right)/3⌋;
                third2= ⌊2*( left + right)/3⌋;

                if key == A[third1]
                        return third1;

                else if key == A[third2]
                        return third2;

                else if key < A[third1]
                        right =ind1 -1;

                else
                {
                        if key > A[third2]
                                left = third2+1

                        else
                        {
                                left = third1+1
                                right = third2-1
                        }
        }
        return 0
}
```

<u>Exercise 4 (10 points)</u>

Give a recursive algorithm for counting number of nodes in a ternary tree. Remember, a ternary tree is a tree such that each node has at most 3 children

```
procedure countNodes(node)
```

```
        if node == NIL
                return 0;
        else
                return 1+ countNodes(node.left) + countNodes(node.mid) +
countNodes(node.right)
```

Exercise 5 (10 points)

Write a recursive algorithm for calculating the product of x and y, making use of the fact that:

- x*y = 3 * (x * y/3) when y is divisible by 3,
- x*y = 3 * (x * (y-1)/3) + x when y -1 is divisible by 3,
- and x * y = 3 *(x * (y-2)/3) + 2x when y - 2 is divisible by 3

```
procedure product (x, y)
        if y%3 = 0
                return 3 * product(x, y/3)
        else if y %3 = 1
                return 3 * product(x, (y-1)/3) + x
        else if y %3 = 2
                return 3 * product(x, (y-2)/3) + 2* x
```

Exercise 6 (10 points)

Devise a recursive algorithm for computing the greatest common divisor of two nonnegative integers a and b using the fact that

$$gcd(a,b) = \begin{cases} gcd(a, b - a) & if\ a < b \\ a & if\ a = b \end{cases}$$

```
Procedure gcd(a,b)
        If a = b
                return a
        else if a < b
                return gcd(a, b-a)
        else
                return gcd (b, a-b)
```

Exercise 7 (10 points)

Devise a recursive algorithm for printing all possible bit strings of length less than or equal to a given integer n.

```
Procedure PrintAll(int n)
        for i = 1 : n
```

```
        PrintStringLen(i, “”)

Procedure PrintLen(int length, String prefix)
    If(length == 0)
        Print(prefix)
    Else
        printLen(i-1, “0” + prefix);
        printLen(i-1, “1” + prefix);
```

Exercise 8 (10 points)

Write a divide and conquer algorithm for finding the max of a list of numbers. What is the time-complexity of your algorithm?

```
Procedure DivNConq_MAX(int list : a1 a2 ... an)
    if n == 1
        return an
    else
        max1 := DIVNConq_MAX (a1 a2 ... an/2)
        max2 := DIVNConq_MAX (an/2+1 a n/2+2 ... an)
        if max1>max2
            return max1
        else
            return max2;
```

$T(n) = 2*T(n/2) + O(1) = O(n)$

Exercise 9 (15 points)

You are given an array of n elements. Devise a divide and conquer algorithm to remove all duplicates in array. What is the runtime of your algorithm?

```
Procedure RemoveDuplicatesRec(int list A : a1 a2 ... an)
    If(n = 1)
        Return A
    else
        FST := RemoveDuplicatesRec(a1 a2 ... an/2)
        SCND:= RemoveDuplicatesRec (an/2+1 a n/2+2 ... an)
        Merged = {}
        i := 1;
        j := 1;
```

```
while(i <= len(FST) AND j <= len(SCND))
        if (FST_i < SCND_j)
                Merged ← FST_i
                i++;
        else if (SCND_j < FST_i )
                Merged ← SCND_j
                j++;
        else
                Merged ← FST_i
                i++;
                j++;

while(i <= len(FST))
        Merged ← FST_i
        i++;

while(j <= len(SCND))
        Merged ← SCND_j
        j++;

return Merged;
```

$T(n) = 2*T(n/2) + O(n) = O(n\lg n)$

Exercise 10 **(10 points)**

Use merge sort to sort 15, 2, 5, 10, 7, 4, 6, 1 into increasing order. Show all the steps used

[15, 2, 5, 10, 7, 4, 6, 1]
[2, 15, 5, 10, 7, 4, 6, 1]
[2, 5, 10, 15, 7, 4, 6, 1]
[2, 5, 10, 15, 4, 7, 6, 1]
[2, 5, 10, 15, 4, 7, 1, 6]
[2, 5, 10, 15, 1, 4, 6, 7]
[1, 2, 4, 5, 6, 7, 10, 15]

A tree representation would be also great

Exercise 11 **(10 points)**

Prove that the merge sort algorithm given in the lecture is correct.

We use strong induction on n, showing that the algorithm works correctly if n = 1, and that if it works correctly for n = 1 through n = k, then it also works correctly for n = k + 1. If n = 1, then the algorithm does nothing, which is correct, since a list with one element is already sorted. If n = k + 1, then the list is split into two lists, L1 and L2 . By the inductive hypothesis, mergesort correctly sorts L1. Now assume that L2 was split into two sublists, the first containing the elements until k and the second contains the (k+1)th element. We know the first sublist would also be correctly sorted using our algorithm given the induction hypothesis, and we know the second sublist which contains only the (k+1)th element is also sorted by definition. So it remains to only show that merge correctly merges two sorted lists into one. This is clear, since with each comparison, the smallest element in $L1 \cup L2$ not yet put into L is put there

Exercise 12 (10 points)

The sum-product of 2 lists is the sum of the product of each term in the first list, with the matching element at same index in the other list. To illustrate, if A = {1, 2, 3} & B = {4, 5, 6}, then AxB = 1*4 + 2 * 5 + 3 * 6 = 32.

a) Write a divide-and-conquer algorithm to calculate the sum-product of 2 lists of same length, by splitting each list into 2 equal (or almost equal) lists, and calling sum-product on each of those sublists, then adding up the total result.

```
Procedure Sum_Product(int list A : a1 a2 ... an, int list B : b1 b2 ... bn)
{
        If n = 1
                return a1 * b1
        else
                FirstSum = Sum_Product(a1 a2 ... an/2, b1 b2 ... bn/2)
                ScndSum = Sum_Product(an/2+1 a n/2+2 ... bn, bn/2+1 b n/2+2 ... bn)
                return FirstSum + ScndSum
}
```

b) What is the recurrence relation resulting from the divide and conquer algorithm defined in a?
$T(n) = 2 * T(n/2) + O(1)$

c) What is the complexity of this algorithm? (use Master's Theorem)
$T(n) = O(n)$

d) Some other programmer decided to split the 2 lists into 3 sublists each instead of 2, and then call he sum-product procedure on those 3 sublists, in order to reduce the number of recursive calls before reaching the base case. What would the recurrence relation be for his algorithm now?

$T(n) = 3*T(n/3) + O(1) = O(n)$

e) Will the algorithm in d have a better run-time complexity compared to that in 1?
Nopes.

Exercise 13 (10 points)

Write the recurrence relation describing the runtime of each of the following recursive algorithms

a) int **rec_ternary_search(int list : a_1 a_2 … a_n , int key, int left, int right)**:

```
{
        If left > right, then return 0

        third1= ⌊ (left + right)/3⌋;
        third2= ⌊2*( left + right)/3⌋;

        if key == A[third1]
                return third1;

        else if key == A[third2]
                return third2;

        else if key < A[third1]
                right =ind1 -1;

        else
        {
                if key > A[third2]
                        left = third2+1

        else
        {
                left = third1+1
                right = third2-1
        }

        return rec_ternary_search(a1 a2 … an , key, left, right);
    }
```

This is usually initially called as on left = 1, right = n

$T(n) = T(n/3) + O(1) = O(\log_3 n)$

b) rec_linear_search(int list : $a_1\ a_2\ \ldots\ a_n$, int key, int index)

```
If key == 0
        Return 0;
Else if key == a_index
    return index
Else
    return rec_linear_search(a1 a2 ... an , key, int index-1)
```

This is initially called on index = n

$T(n) = T(n-1) + O(1) = O(n)$

c) MAXMIN(list S : $a_1, a_2, \ldots, a_n$)

```
if n = 2 then
    if a1 ≥ a2 then
            return (a1, a2)
    else
            return (a2, a1)
else if n == 1
    return (a1, a1)

S1 ← {a1, a2, . . . , a_{n/4}}
S2 ← {a_{n/4+1}, a_{n/4+2}, . . . , a_{n/2}}
S3 ← {a_{n/2+1}, a_{n/2+2}, . . . , a_{3n/4}}
S4 ← {a_{3n/4+1}, a_{3n/4+2}, . . . , a_n}

(max1, min1) ← MAXMIN(S1)
(max2, min2) ← MAXMIN(S2)
(max3, min3) ← MAXMIN(S3)
(max4, min4) ← MAXMIN(S4)

maxs = [max1, max2, max3, max4]
mins = [min1, min2, min3, min4]

(max, dummy) ← MAXMIN(maxs)
(dummy, min) ← MAXMIN(mins)

return (max, min)
```

$T(n) = 4*T(n/4) + O(1) = O(n)$

Exercise 14 **(10 points)**

Use Master's theorem to complexity of each of the following recurrences

a) $T(n) = 2T(n/2) + n^2$
   $T(n) = \Theta(n^2)$

b) $T(n) = T(n/2) + O(1)$
   $T(n) = \Theta(\lg n)$

c) $T(n) = 2T(n/2) + O(1)$
   $T(n) = \Theta(n)$

d) $T(n) = 4T(n/2) + O(n^3)$
   $T(n) = \Theta(n^3)$

e) $T(n) = 2T(n/2) + 7n$
   $T(n) = \Theta(n \lg n)$

Please solve the following exercises and submit **BEFORE 11:55 pm of Sunday 22nd of November.**

Please get a hardcopy submission whether you plan to solve it on a computer or on a paper. In Both cases, also **submit to Moodle**. However, if you submit a handwritten solution, I will only correct the questions that I manage to read (and easily find).

Pass the hardcopy submissions during the week, starting from Monday 23rd . I will use Moodle submissions as a proof of early submissions. Don't try to modify anything in the hard copy submission, or else it will be considered cheating and you'll get a Zero.

Exercise 1 (15 points)

A bowl contains 10 red apples and 10 yellow apples. A man selects apples at random without looking.

a) How many apples must he select to be sure of having at least three apples of same color?
There are 2 colors. Those are pigeonholes. We want ceil(N/2) = 3, then N = (3-1)*2 +1 = 5

b) How many apples must he select to be sure of having at least 3 yellow apples?
In worst case, he will pick all red apples first, then the yellow ones. Therefore he needs to pick 10 + 3 = 13 apples to ensure that 3 yellow apples have been picked.

Exercise 2 (15 points)

How many bit strings of length 10 contain

a) Exactly four 1s?
Simply, We need to choose 4 positions containing 1s; 10C4 = 210

b) At most four 1s?
Having at most four means having zero 1s, having one 1s, two, three or four 1s. Therefore, 10C0 + 10C1 + 10C2 + 10C3 + 10C4 = 386

c) At least four 1s?
Same as part b, we get 848

Another approach is calculating all possible bitstrign of length 10, and subtracting from them those bit strings not having at least four 1s,
And therefore 2^10 – 10C0 – 10C1 – 10C2 -10C3 = 848 also

d) An equal number of 0s and 1s?
To have an equal number of 0s and 1s means having five 1s. Therefore 10C5 = 252.
However, this can lead to another way for doing part b; if we don't have an equal number of 1s and 0s, then we have either at most four 1s or at least 6 ones, By symmetry, having at most four 1s occurs in half of these cases.
Therefore, the anwer of part b is (2^10 – 10C5)/2 = 386, as shown in part b

e) Either 5 consecutive 0s or 5 consecutive 1s
First we count the number of bit strings of length 10 that contain five consecutive 0's. We will base the count on where the string of five or more consecutive 0's starts. If it starts in the first bit, then the first five bits are all 0's, but there is free choice for the last five bits; therefore there are 25 = 32 such strings. If it starts in the second bit, then the first bit must be a 1, the next five bits are all 0's, but there is free choice for the last four bits; therefore there are 24 = 16 such strings. If it starts in the third bit, then the second bit must be a 1 but the first bit and the last three bits are arbitrary; therefore there are 24 = 16 such strings. Similarly, there are 16 such strings that have the consecutive 0's starting in each of positions four, five, and six. This gives us a total of 32 + 5 * 16 = 112 strings that contain five consecutive 0's. Symmetrically, there are 112 strings that contain five consecutive 1's. Clearly there are exactly two strings that contain both (0000011111 and 1111100000). Therefore by the inclusion–exclusion principle, the answer is 112 + 112 − 2 = 222.

Exercise 3 (15 points)

How many string of six lowercase letter from the English alphabet contain

a) The letter a?
The only reasonable way to do this is by subtracting from the number of possible strings of length 6 the number of strings of length 6 with no a in them, and therefore 26^6 – 25^6 = 64775151

b) The letters a and b?
If out string is to contain both of these letters, then we need to subtract from the total number of strings the number that fails to contain one or the other or both of these letters. As in part (a), 25^6 strings fail to contain the letter a,

similarly 25^6 fail to contain b. This is overcounting; however, 24^6 fail to contain both a and b. Therefore 25^6 + 25^6 -24^6 fail to contain either a or b (or both). Therefore, 26^6 – (25^6 + 25^6 -24^6) is the answer, 11,737,502.

c) The letters a and b in consecutive positions, with a preceding b, with all the letters distinct?
First choose the position for the a; this can be done in 5 ways since be most follow it. There are four remaining positions, and these can be filled in 24P4 ways, since there are 24 letters left. Therefore we have 5 * 24P4 = 1,275,120 ways

d) The letters a and b, where a is somewhere to the left of b in the string, with all the letters distinct?
First we choose the 2 positions out of the 6 possible positions, for the a and the b; This can be done in 6C2 ways, since once we pick two positions, we put the a in the left-most and the b in the other. There are four remaining positions, and these can be filled in 24P4 ways, since there are 24 letters left (no repetitions being allowed this time). Therefore the answer is 6C2* 24P4 = 3,852,360

Exercise 4 (15 points)

How many permutations of the digits 0, 1, … 9, starts with 3 **or** ends with 7?

a) If duplicate digits not allowed (10 digit number)
9! + 9! – 8! = 685440

b) If duplicate digits allowed (10 digit number)
10^9 + 10^9 –10^8 = 1,900,000,000

c) If duplicate digits allowed, but the digits are in non-decreasing order, and we need a 3 digit number only
Start with 3: 1 + 2 + 3 + 4 + 5 + 6 + 7 = 28
Ends with 7: 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36
Starts with 3 and ends with 7 = 5
Result = 28 + 36 – 7 = 59

Exercise 5 (15 points)

Show that if there are 30 students in a class, then at least two have last names that begin with the same letter.

Ceil(30/26) = 2

Exercise 6 (15 points)

A group contains n men and n women. How many ways are there to arrange these people in a row if the men and women alternate?

N! * N!

Exercise 1 (15 points)

Give an example of two countable sets A and B such that A – B is

a) Finite
A = R, B=R-0, A-B = empty, which is finite

b) Countable infinite
A = R, B = R – Z, A –B = Z, which is countably infinite

c) Uncountable
A= R, B =R+, then A-B = R-

Exercise 1 (15 points)

Give an example of two countable sets A and B such that A ∩ B is

a) Finite
R+, B=R-, A ∩ B = empty, which is finite

b) Countable infinite
A= (2,3) U Z+, B = (4,5) U Z+, A ∩ B = Z+

c) Uncountable
A= (2,4), B=(3,4), A ∩ B = (3,4)

Exercise 1 (15 points)

Determine whether each of these sets is finite, countably infinite, or uncountable. For those that are countably infinite, exhibit a one-to-one correspondence between the set of positive integers and that set.

a) the even integers

C, F(i) = i if i is odd, and –i-1 if i is even

b) the real numbers between 0 and 1
U

c) the positive integers less than 1,000,000,000
F

d) the integers with absolute value less than 1,000,000
F

e) the real numbers between 0 and 2
U

f) the set A × Z+ where A = {1, 7}
C

g) all positive rational numbers that cannot be written with denominators less than 4
C

h) the real numbers not containing 0 in their decimal representation
U

i) the real numbers containing only a finite number of 1s in their decimal representation
C

j) integers not divisible by 3
C

k) the real numbers with decimal representations of all 1s or 9s
U

Exercise 1 (15 points)

For each of those countably infinite, exhibit a one-to-one correspondence between the set of positive integers and that set.

a) the integers that are multiples of 7
C, F(i) = 7i/2 if n is even, and -7(i-1)/2 if i is odd

b) all bit strings not containing the bit 0

C

c) the set A × Z+ where A = {2, 3}
C, F(i, j) = (2, (i-1)/2) if i is odd, (3, i/2 -1) if i is even

d) integers divisible by 5 but not by 7
C

Exercise 1 (15 points)

Use a loop invariant to prove that the following program segment for computing the nth power, where n is a positive integer, of a real number x is correct.

```
Procedure power(int X, int N)
    power := 1
    i := 1
    while i ≤ n
        power := power * x
        i := i + 1

    return power;
```

```
Procedure power(int X, int N)
    //power = x^0;
    power := 1
    i := 1
    //power = x^(i-1) And i = 1;
    while i ≤ n
        //power = x^(i-1) And i <= n;
        power := power * x
        //power = x^i And i <= n;
        i := i + 1
        //power = x^(i-1) And i <= n+1;

    //power = x^i-1 AND i <= n+1 AND I >n
    //→ power = x^i-1 AND i = n+1
    //→ power = x^n
```

return power;

Exercise 1 (15 points)

Prove that the set of all polynomials of degree <= 3 with integer coefficients is countable. [Hint: use the proof that Q is countable shown in the slides]

Let P3 be the set of all all polynomials of degree <= 3 with integer coefficients. Thus
P3 = {a0 + a1x +a2x^2 + a3x^3 | a0, a1, a2, a3 are integers)
Since the set Z is countable, the result of Problem 3 above implies that Z^4 = Z x Z x Z x Z is also countable.
Define a function f: Z^4 → P3 by the formula:

F(a0, a1, a2, a3) = a0 + a1x +a2x^2 + a3x^3

Where a1, a2, a3 belong to Z. By construction the function f is surjective.
Hence the statement (3) above implies that P3 is countable.

Exercise 1 (15 points)

There are 50 baskets of apples. Each basket contains no more than 24 apples. Show that there are at least 3 baskets containing the same numbers of apples.

Ceil(50/24) = 3

Exercise 1 (15 points)

AUB faculty emails are formed of 2 letters (this initials of the faculty member) followed by a number. What is the least possible of faculty members that need to work at AUB to have all possible email initials covered? (Think of the best case, where each new faculty member has a distinct initial than all old members)

26^2

Exercise 1 (15 points)

A coin is flipped 10 times, where each flip comes up either heads or tails. How many possible outcomes

a) Are there in total?
   2^10 = 1024

b) Contain exactly two heads?
   10C2 = 45

c) Contain at most three tails?

$0C10 + 1C10 + 2C10 + 3C10 = 176$

d) Contain the same number of heads and tails?
$5C10 = 252$

Exercise 1 (15 points)

A club has 25 members

a) How many ways are there to choose four members of the club to serve on an executive committee?
$25C4 = 12650$

b) How many ways are there to choose a president, vice president, secretary, and treasurer of the club, where no person can hold more than one office?
$25P4 = 303600$

Exercise 1 (15 points)

How many ways are there for a horse race with three horses to finish if ties are possible? [Note: two or three horses may tie]

$3! + 3 + 3 + 1 = 13$

Exercise 1 (15 points)

How many ways are there for a horse race with four horses to finish if ties are possible? [Note: two or three or four horses may tie]

$4! + 4*3 + 4 \ + 4 * 3 + 4 * 3 + 4*3/2 + 4 + 1 = 24 + 12 + 4 + 12 + 12 + 6 + 4 + 1 = 75$

Exercise 1 (15 points)

25 Horses are made to race with each other to find out the fastest 3. Assume all horses have distinct speeds, and any horse will perform exactly the same, regardless of how many times it races (doesn't get tired/slows down/or speeds up). If you were to race 5 horses in each race, what is the minimum number of races needed to find out which 3 horses are the fastest?

5 Races to race all horse
1 Race between the first from the 5 races
1 Race between 2,3 from Race 2, and 2,3 from the race the fastest horse was in, and 2 from the race the 2nd fastest horse was in
Total 7

Exercise 1 (15 points)

A vendor sells ice cream from a cart on the boardwalk. He offers vanilla, chocolate, strawberry, and pistachio ice cream, served on either a waffle, sugar, or plain cone. How many different tripe-scoop ice-cream cones can you buy from this vendor? (choice of same flavor is allowed)

$4^3 * 3 = 192$

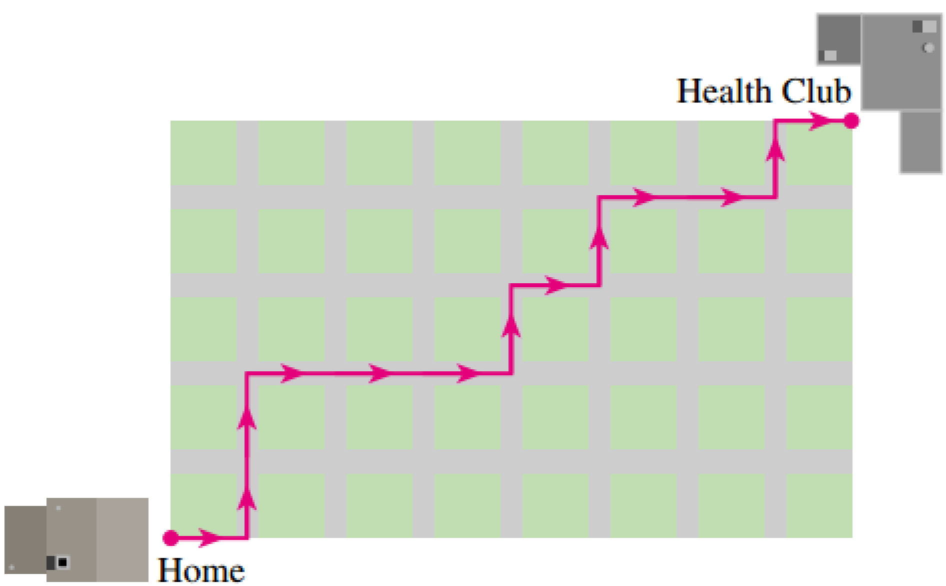
Exercise 1 (15 points)

A Health club member goes daily from his home to a nearby Health club by crossing roads either in the East or North directions. He goes neither West nor South.

If his home is on one corner, and the health club is on the opposite corner of a 3-by-3 square street grid, how many days does he need in order to go through all possible routes from home till health clubs, if he goes in a different route each day?

Hint: Count the number of ways in an iterative manner, moving from bottom left, and expanding the path 1 step at a time. As in, start from (0,0) – Home-, and count the number of ways you can reach (0,1) and (1,0), and then deduce (1,1)..and so on till position (n,n) – which is the Health Club in this case

You may also think about it as a combination of East and North moves.



American University of Beirut
Department of Computer Science
CMPS 211 – Discrete Mathematics – Fall 15/16
Assignment 8

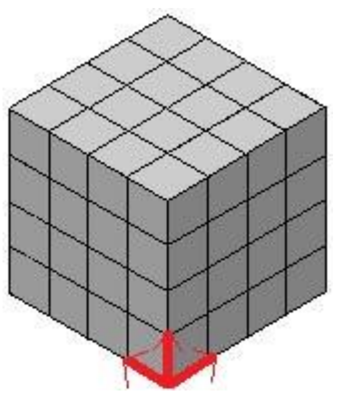
20

Exercise 1 (15 points)

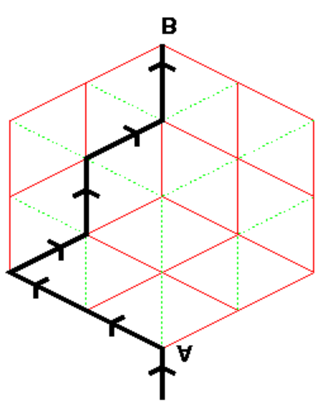
You are given a 4-by-4-by-4 cube. You need to go from position (0,0,0) to (4,4,4). You are allowed to go on the edges of the subcubes, traversing from one subcube to another, but you aren't allowed to pass inside a subcube. ; i.e.: if you are on (1,2,2) you can go to (1,2,3) or (1,3,2) or (2,2,2); you aren't allowed to go backwards. Check figures below to better understand the problem. *If you want, implement it to get the answer*

a) In how many ways can you reach your destination? (This is the same as the previous problem but in 3D Cube instead of 2D Square)
34650

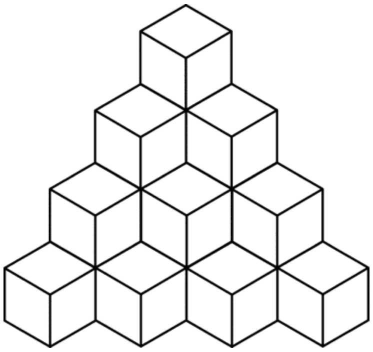
b) Can you find a generalized equation for any cube of side n?
$(n + n + n)!/(n! * n! * n!)$



Start point, and allowed moves



Example Route



Cube dissection, You are allowed to pass through those edges

Exercise 1 (15 points)

A five-person committee consisting of students and teachers is being formed. 12 Teachers and 14 students have been selected as candidates for the committee. The rules state that at least 1 teacher and 1 student should be in the committee. In how many ways can this committee be formed?

12 * 14 * 24C3 = 340,032

Exercise 1 (15 points)

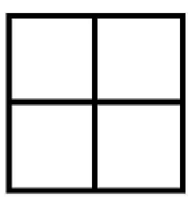
How many different ways are there to mark the answers to a 20-question multiple choice test in which each question has 4 possible answers?

4^20 = 1,099,511,627,776

Exercise 1 (15 points)

Prove that among any five points selected inside a square with side length 2 units, there always a pair of these points that are with sqrt(2) units of each other?

We can argue using the pigeon-hole principle…
Suppose we divide our square into 4 smaler square regions each of side length 1 unit, like so:



These smaller square regions will be our "pigeonholes". Now, if one selects 5 points (our "pigeons") inside the larger square, and noting that 5 is one more than the number of smaller square regions – the pigeon hole principle requires that 2 of these points be in at least one of the smaller squares. Since the maximum distance between two points in one of these smaller squares is the length of the diagonal of that square, namely sqrt(2), we know there exist a pair of points out of the group of five originally selected that will be within sqrt(2) units of each other.

Exercise 1 (15 points)

Rearrange the first n integers: 1,2,3, …, n listing them as a1, a2, … an. Prove that if n is odd, then (a1 – 1)(a2 – 2) … (an – n) must be even, regardless of how the integers were initially rearranged.

We argue indirectly…
Assume (a1 – 1) (a2 – 2) … (an – n) is odd. Then each factor (a1 – 1), (a2 – 2), … (an – n) must be odd. This implies that a1, a3, a5, … , an are all even, while a2, a4, … an-1 are all odd.
Contemplating the potential use of the pigeon-hole principle (although we end up not use it after all), we count the number of odds and find there exists (n-1)/2 of the ai's that are odd.
However, the numbers a1, a2, … , an are precisely 1, 2, 3, … n , just possibly in a different oder. The number of odd numbers in this last list is (n-1)/2 +1, so we have a contradiction.
Hence, we must reject our initial assumption that the product (a1 – 1)(a2 -2) … (an – n) could odd. Instead, it must be even.

Exercise 1 (15 points)

What is the largest number of kings which can be placed on a chessboard so that no two of them put each other in check? A chessboard is an 8-by-8 board, and a king can move 1 step in all possible directions (Draw it).

16

Exercise 1 (15 points)

Prove that among any 52 integers, two can always be found such that the difference of their squares is divisible by 100.

Hint: Use pigeon hole principle, and express the numbers in a form n = pq + r as when calculating remainders.

Any number can be expressed as n = 50k + r, where 0<=r<50.
Since we are picking 52 integers, then by pigeon hole principle at least 2 integers ni = 50ki + ri and nj = 50kj + rj, such that ri = rj

Without loss of generality, pick those 2 integers (n1 and n2) that have equal values of r, such that n1 > n2
Then n1^2 – n2^2 = (50k1 + r)^2 – (50k2 + r)^2
= 2500k1^2 + r^2 + 100k1*r - 2500k2^2 - r^2 - 100k2*r

= 2500(k1^2 – k2^2) + 100 r (k1 – k2)
= 100 (25 (k1^2 – k2^2) + r (k1 – k2))
= 100 m, then n1^2 – n2^2 is a multiple of 100, and divisible by 100

Exercise 1 (15 points)

Write the post-condition of the following procedure, that calculate the absolute value of a number

```
Procedure abs( n )
    int absolute = 0;
    if n < 0
        absolute = -n
    else
        absolute = n

    return absolute
```

```
Procedure abs( n )
int absolute = 0;
if n < 0
    //n < 0
    absolute = -n
    //absolute = -n and n < 0
    //absolute = |n|
else
    // n >= 0
    absolute = n
    //absolute = n and n >= 0
    //absolute = |n|

//Post-Condition: (absolute = -n AND n < 0) OR (absolute = n AND n >= 0)
//absolute = |n|
return absolute
```