Problem 1.(15 Points)

The following is the algorithm for sorting by insertion

```
procedure insertion sort (a1,a2,...,an: real numbers with n ≥ 2)
1    for j := 2 to n
2    begin
3         position := 1
4         while aj > aposition
5              position := position + 1
6         m  : = aj
7         for k := 0 to j – position - 1
8              aj-k := aj-k-1
9          aposition  : = m
10   end { a1,a2,...,an  are sorted}
```

For each j, the algorithm does the following: $a_1,\dots,a_{j-1}$ is already sorted, it inserts a_j in the "correct position" so that $a_1,\dots,a_j$ is sorted. The "correct position" is computed in lines 3-4-5 in the variable i. It is "searched" for in a way based on linear search.

a. (4) Complete lines 5 and 8 above

b. (4) Explain why the while loop in line 4 must terminate (i.e. its condition will definitely become false.)

   Eventually, position will become j and so $a_j = a_{position}$ (worst case)

c. (4) Use the insertion sort procedure above to sort the list 6, 2, 3, 1, 4 showing the lists obtained at the end of each step. Here n=5. Fill out the following table.

j	a_1	a_2	a_3	a_4	a_5
	6	2	3	1	4
2	2	6	3	1	4
3	2	3	6	1	4
4	1	2	3	6	4
5	1	2	3	4	6

d. (3) For the step j = 3, what will be the value of i in line at the end of the loop of lines 4-5, and what is the stopping condition that will terminate the loop?

   $i = 2$

   Stopping condition : 3 > 6 False

Problem 2.(15 Points)
This problem refers to the insertion sort algorithm given in the previous problem. You are supposed to do a **worst-case** complexity analysis for insertion sort, based on the count of number of comparisons used. Assume that in the **for** iterations in lines 1 and 7, one comparison is used for each loop, to check whether the "limit has been reached".

a. (4) For each j how many comparisons $C_1(j)$ will be used to execute lines 3-4-5 ?

$C_1(j)$ = position + 1 (position < j)

b. (4) For each j how many comparisons $C_2(j)$ will be used to execute lines 6-7-8-9 ?

$C_2(j)$ = j – position + 1

c. (4) What is the overall count $C(n)$ of comparisons used to sort a list with n elements? Show your computations.

$$C(n) = \sum_{j=2}^{n}(C_1(j) + C_2(j) + 1) + 1$$
$$= \sum_{j=2}^{n}(position + j - position + 2) + 1$$

$$C(n) = \frac{n(n+1)}{2} - 1 + 2(n-1) + 1$$
$$= \frac{n^2}{2} + \frac{5n}{2} - 2$$

d. (3) Based on your answer in part (c), what is the big-O estimate that you can give for the algorithm insertion sort ? { Use **only one** of *log n, n, n log n,* n^2 , n^3}

$C(n) = O(\quad n^2 \quad)$

Problem 3.(10 Points)
The following algorithm is based on binary search for determining the correct position in which to insert a new element x in an already sorted list.

procedure *binary search position* (x : integer, $a_1, a_2, \ldots, a_n$: increasing integers)
    $i := 1$ {i is left endpoint of search interval}
    $j := n$ {j is right endpoint of search interval}
    while $i < j$
        begin
            $m := \lfloor (i+j)/2 \rfloor$
            if $x > a_m$ **then** $i := m+1$ **else** $j := m$
        end
    if $x > a_i$ **then** $position := i + 1$
    else $position := i$
{*position* is the correct position to insert x }

a. (2) Complete the last **else** line of the algorithm above.

b. (4)What is the count of comparisons for this algorithm? Why?

Let k be such that $2^k \leq n < 2^{k+1}$ i.e. $k = \lfloor Log_2 n \rfloor$

Number of comparisons = 2k + 1 (for the while loop)
+ 1 (for the if)
= 2k + 2 = $2Log_2 n + 2$

c. (4) The algorithm above will be used in the insertion sort algorithm above for determining the correct position in which to insert a_j in the list $a_1, a_2, \ldots, a_{j-1}$. So lines 3-4-5 in *insertion sort*, will be replaced by a "call" to the binary search position procedure as follows:

binary search position (a_j , $a_1, a_2, \ldots, a_{j-1}$)

How would your answers to parts (c) and (d) of the previous problem change?

C(n) = $[\sum_{j=2}^{n} (2Log_2(j-1) + 2 + j - position + 1] + 1$

position = 1 worst case

$C(n) = (2 \sum_{j=2}^{n} Log_2 (j-1) + 2 + j) + 1$

$C(n) = O($ n^2 $)$ { Use only one of $log\ n$, n, $n\ log\ n$, n^2 , n^3}

Problem 4.(5 Points)

a. Use the definition of big-oh to prove that $1.2 + 2.3 + ... + (n\text{-}1).n$ is $O(n^3)$.

$1.2 + 2.3 + \dots + (n\text{-}1).n \leq (n\text{-}1).n + (n\text{-}1).n + \dots + (n\text{-}1).n$
[n-1 times]

$= (n\text{-}1)^2.n \leq n^3$ for all $n \geq 2$

Which is $O(n^3)$; witnesses $c = 1$ and $n \geq 2$

Problem 5. (15 Points)

Consider the proposition

P(n): An amount of postage of *n* cents can be formed using 3-cent and 7-cent stamps.

You should prove that *P(n) is true* for $n \geq 12$, first using mathematical induction and then using strong mathematical induction.

a. (8) Use mathematical induction to prove that *P(n) is true* for $n \geq 12$.

P(12) = 4*3 + 0*7 true

Assume it is true for k ≥ 12. Show it for P(k+1)

k = a*3 + b*7

if a≥ 2, then k + 1 = a*3 + b*7 + 1
= (a-2)*3 + (b+1)*7

if a = 0, then k = 7b and since k ≥ 12 ⇒ b ≥ 2
k + 1 = 7*(b-2) + 15 = 5*3 + (b-2)*7

if a = 1, then k = 3 + 7*b and again since k ≥ 12, then b ≥ 2
so k + 1 = 6*3 + (b-2)*7

therefore P(n) is true for all n ≥ 12.

b. (7) Use strong mathematical induction to prove the result. (Hint: Show that the statements *P*(12), *P*(13), and *P*(14), are true, and use strong induction accordingly)

P(12) = 4*3 + 0*7, true
P(13) = 2*3 + 1*7, true
P(14) = 0*3 + 2*7

Assume p(k) is true for k = 12, 13, 14, …, n ; n ≥ 14

Consider n+1
n+1 - 3≥ 14 + 1 – 3 ≥ 12
so P(n+1-3) is true
n + 1 – 3 = a*3 + b*7
n + 1 = (a+1)*3 + b*7, then P(n+1) is true

therefore P(n) is true for all n ≥ 12

Problem 6. (10 Points)
In the questions below give a recursive definition with initial condition(s).

a. (4) The function $f(n) = 2^n$, $n = 1, 2, 3, ...$

Basis step: $f(1) = 2$

Recursive step: $f(n) = 2*f(n-1)$, $n \geq 2$

b. (3) The sequence $a_1 = 16$, $a_2 = 13$, $a_3 = 10$, $a_4 = 7$, ...

Basis step: $a_1 = 16$

Recursive step: $a_n = a_{n-1} - 3$, $n \geq 2$

c. (3) The set $\{0,3,6,9,...\}$.

Basis step: $0 \in S$

Recursive step: $x \in S \rightarrow x + 3 \in S$

Problem 7.(15 Points)

In the questions below suppose that a "word" is any string of seven letters of the English alphabet, with repeated letters allowed. There are 26 letters in the English alphabet, and there are 5 vowels (a, e, i, o, u.) Show your computations

a. (2) How many words are there?

26^7

b. (2) How many words begin with R **and** end with T?

26^5

c. (3) How many words begin with A **or** end with B?

Begin with 'A': 266
End with 'B': 266
Begin with 'A' and end with 'B': 265
Begin with 'A' or end with 'B': 2 * 266 - 265

d. (4) How many words begin with the first three letters are A, A, B in some (any) order?

A, A, B in some order: 3 possibilities: AAB, ABA, BAA

So 3 * 264

e. (4) How many words **have exactly** 2 vowels?

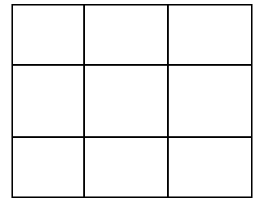
Number of possibilities for the position of these 2 vowels =

$$\binom{7}{2} = \frac{7!}{2!5!} = \frac{7!}{2!5!} = \frac{7*6}{2!} = 21$$

For each possibility: 215
So total: 21 * 25 * 215

Problem 8 (15 Points)

a. (5) Show that if 10 points are picked on or in the interior of a square of side length 3, then there are at least two of these points no farther than $\sqrt{2}$ apart.



The square can be divided into 9 squares each of dimensions 1 x 1.
2 of the 10 points have to be in one of these squares (by pigeon hole principle)
The distance between these two points is at most $\sqrt{2}$ (diagonal distance – max in square)

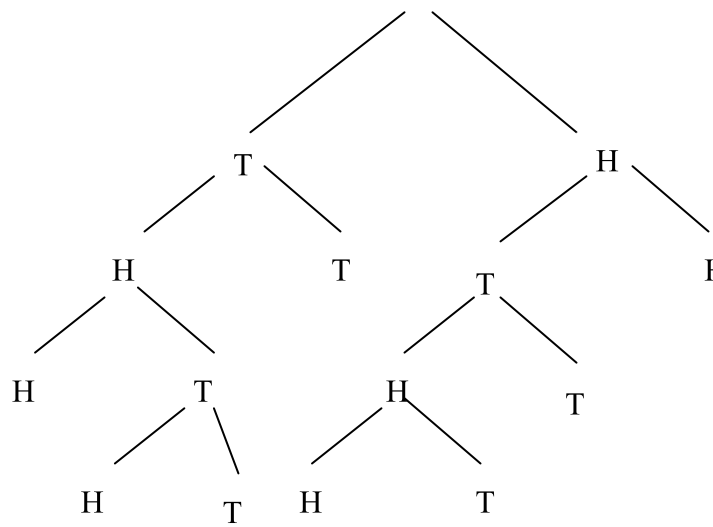
b. (5) Find the least number of cables needed to connect 10 computers to 4 printers to guarantee that 6 computers can directly access 4 different printers. Justify your answer.

If 30 cables are given, then they can be used so that each of the 10 computers get connected to 3 printers each.
To guarantee that 6 computers are connected to 4 different printers each: 36 cables

c. (5) A game consisting of flipping a coin ends when the player gets two heads in a row, two tails in a row, or flips the coin four times.

(a) Draw a tree diagram to show the ways in which the game can end.

(b) In how many ways can the game end?



8 ways

Problem 9. (10 Points)

a. (5) How many functions are there from the set $\{1, 2, \ldots, n\}$ to the set $\{1, 2, 3\}$. Why?

For each of 1, 2, …, n there are 3 possible images $\Rightarrow 3^n$

b. (5) Out of the functions in (a), how many are one-to-one? Why?

If $n > 3$, then 0 functions

If $n \leq 3$, then:

<u>n = 3:</u>

for 1 we have 3 choices
for 2 we have 2 choices
for 3 we have 1 choice

3 * 2 * 1 = 6 choices

<u>n = 2:</u> 3 * 2

<u>n = 1:</u> 3

n	3 … (3 – n + 1)
3	3*2*1
2	3*2
1	3