AUB-CMPS 211-Spring 2015-16	March 29th, 2016

Quiz 2

Instructors: F. Abu Salem and A. Al Zaatari **Name:**

Duration: 1 hour

This exam is closed notes.

Question 1 (15%)

Let $A = \{1, 2, 3\}$ and $B = \{1, 2, 5, 10\}$.

(a) (5%) Find $A \cup B$ and $A \cap \overline{B}$. (The answer does not depend on the choice of universal set)
Solution:

(a) $A \cup B = \{x \mid (x \in A) \vee (x \in B)\} = \{1, 2, 3, 5, 10\}$. 2.5 pts

$A \cap \overline{B} = \{x \mid (x \in A) \wedge (x \notin B)\} = \{3\}$. 2.5 pts

(c) $\forall x \in A\ \exists y \in B\ (x|y)$.
Alternatively: $\forall x \in A\ \exists y \in B\ \exists n \in \mathbf{Z}\ (y = nx)$. 2.5 pts
This statement is false since $3 \in A$ and $\neg(3|1) \wedge \neg(3|2) \wedge \neg(3|5) \wedge \neg(3|10)$. 2.5 pts

(d) $\exists x \in A\ \forall y \in B\ x|y$. 2.5 pts
This statement is true since $1 \in A$ and $(1|1) \wedge (1|2) \wedge (1|5) \wedge (1|10)$. 2.5 pts

Question 2 (15%)

Let $f : \mathbf{Z} \to \mathbf{Z}$ be the function defined by $f(n) = \left\lceil \frac{n}{3} \right\rceil$.
Solution:

(a) $f(1) = 1$, $f(3) = 1$, $f(4) = 2$ and $f(100) = 34$. Hence $f(\{1, 3, 4, 100\}) = \{1, 2, 34\}$. 2.5 pts

(b) $f^{-1}(\{-1, 10\}) = \left\{ n \in \mathbf{Z} \mid \left\lceil \frac{n}{3} \right\rceil \in \{-1, 10\} \right\}$. 1 pt

So for $n \in \mathbf{Z}$, we have (1 pt):

$$\begin{aligned} n &\in f^{-1}(\{-1, 10\}) \\ &\leftrightarrow (\lceil \tfrac{n}{3} \rceil = -1) \vee (\lceil \tfrac{n}{3} \rceil = 10) \\ &\leftrightarrow (-2 < \tfrac{n}{3} \leq -1) \vee (9 < \tfrac{n}{3} \leq 10) \\ &\leftrightarrow (-6 < n \leq -3) \vee (27 < n \leq 30). \end{aligned}$$

This means that $f^{-1}(\{-1, 10\}) = \{-5, -4, -3, 28, 29, 30\}$. 0.5 pts

(c) f is not an injection since $f(1) = f(2) = 1$ even though $1 \neq 2$. 2.5 pts
In other words, there are distinct elements of $\mathbf{Z}$ having the same image by f. 2.5 pts

(d) f is a surjection since for all $n \in \mathbf{Z}$, we can find a preimage $k = 3n$ for which $f(3n) = n$. 2.5 pts
Hence every integer n has at least one preimage. (Note that the other preimages of n are $3n - 1$ and $3n - 2$.) 2.5 pts

Question 3 (20%)

Use **strong induction** to show that every positive integer n can be written as the sum of **distinct** powers of two, that is, as a sum of a subset of integers $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, ..., etc. When attempting to infer $P(k+1)$ from previous instances of that same predicate, consider the following hints:

- Hint 1: Treat the cases when k is even or odd separately.
- Hint 2: When k is odd, use the fact that $\frac{k+1}{2}$ is an integer.

Solution

Base Case: Take $k = 1$ the first positive integer, k can be written as $k = 2^0 = 1$. Therefore $P(k)$ is true. 4 pts

Inductive Step: Assume That the hypothesis holds for all positive integers up to and including k, in other words:

$$\forall i \in \mathbb{N}^* \ (i \leq k) \rightarrow P(i)$$

2 pts

We need to show that $P(k+1)$ holds. We proceed by cases on k.

- if k is even, then $k+1$ is odd. By the *inductive hypothesis*, k can be written as a sum of powers of 2:

  $$k = 2^{m_1} + 2^{m_2} + ... + 2^{m_n}$$

  Therefore we can write $k+1$ as:

  $$k + 1 = 2^{m_1} + 2^{m_2} + ... + 2^{m_n} + 1 = 2^{m_1} + 2^{m_2} + ... + 2^{m_n} + 2^0$$

  Therefore $P(k+1)$ holds. 7 pts

- if k is odd, then $k+1$ is even, therefore $\frac{k+1}{2}$ is a positive integer, and $\frac{k+1}{2} < k+1$. By the *inductive hypothesis*, $\frac{k+1}{2}$ can be written as a sum of powers of 2:

  $$\frac{k+1}{2} = 2^{m_1} + 2^{m_2} + ... + 2^{m_n}$$

  However $k + 1 = 2\frac{k+1}{2}$. Which can be writter as:

  $$k + 1 = 2(2^{m_1} + 2^{m_2} + ... + 2^{m_n}) = 2 \times 2^{m_1} + 2 \times 2^{m_2} + ... + 2 \times 2^{m_n}$$

  $$k + 1 = 2^{m_1+1} + 2^{m_2+1} + ... + 2^{m_n+1}$$

  Therefore $P(k+1)$ holds. 7 pts

Question 4 (15%)

Use mathematical induction to prove that $\sum_{i=1}^{n} \frac{1}{(2i)(2i+2)} = \frac{n}{4(n+1)}$

Solution

Let $P(n) =$ "$\sum_{i=1}^{n} \frac{1}{(2i)(2i+2)} = \frac{n}{4(n+1)}$". 3 pts **Base case:** Take $n = 1$, then:

$$\sum_{i=1}^{n} \frac{1}{(2i)(2i+2)} = \frac{1}{2(2+2)} = \frac{1}{4(1+1)} = \frac{n}{4(n+1)}$$

3 pts

Inductive step: Show $P(n) \to P(n+1)$. Assume that $\sum_{i=1}^{n} \frac{1}{(2i)(2i+2)} = \frac{n}{4(n+1)}$. Write:

$$\begin{aligned}
\sum_{i=1}^{n+1} \frac{1}{(2i)(2i+2)} &= \sum_{i=1}^{n} \frac{1}{(2i)(2i+2)} + \sum_{i=n+1}^{n+1} \frac{1}{(2i)(2i+2)} \qquad \text{3 pts} \\
&= \frac{n}{4(n+1)} + \frac{1}{4(n+1)(n+2)} \qquad \text{3 pts} \\
&= \frac{n(n+2)}{4(n+1)(n+2)} + \frac{1}{4(n+1)(n+2)} \\
&= \frac{n^2+2n+1}{4(n+1)(n+2)} \\
&= \frac{(n+1)^2}{4(n+1)(n+2)} \\
&= \frac{n+1}{4(n+2)} \\
&= \frac{n+1}{4((n+1)+1)} \qquad \text{3 pts}
\end{aligned}$$

Therefore, $P(n+1) =$ "$\sum_{i=1}^{n} \frac{1}{(2i)(2i+2)} = \frac{n}{4(n+1)}$" is established.

Question 5 (35%)

Given a list (tuple) of integers $A = (a_1, \ldots, a_n)$, the pre-fix sum (cumulative sum) of A is another list (tuple) $S = [s_1, \ldots, s_n]$ such that $s_i = a_1 + \ldots + a_i$. For example, given $A = [1, 5, 9, 12, 100]$, we have $S = [1, 6, 15, 27, 127]$.

(a) (10%) Develop the algorithm (in pseudo-code), which, given a list (tuple) of integers A, produces their pre-fix sum S.

Procedure prefixsum(a_1, ..., a_n):
  Let s_1, ..., s_n be the list that will contain the prefix sum.
  $s_1 := a_1$
  for $i := 2$ **to** n
    $s_i := s_{i-1} + a_i$

  return s_1, ..., s_n

(b) (2.5%) Can you run your alogrithm in-place (that is, you produce the output in the same data structure as that hosting your input?)? Explain why or why not.

After iteration i, the entry a_i is no longer needed. This claim is valid for all iterations. So, the algorithm can run in-place, that is, we can procuce the output in the same array as the input.

Procedure inplace-prefixsum(a_1, ..., a_n):
  for $i := 2$ **to** n
    $a_i := a_{i-1} + a_i$

  return a_1, ..., a_n

(c) (5%) State a loop invariant for the algorithm.

At the start of iteration i, we have:

$$s_k = \sum_{j=1}^{k} a_j = a_1 + ... + a_k \quad \forall k < i$$

For the in place version, the invariant becomes:

$$a_k = \sum_{j=1}^{i} a_j = a_1 + ... + a_k \quad \forall k < i$$

(d) (10%) Prove the loop invariant for the algorithm.

Initialization: At the start of the first iteration $i = 2$, we have $s_1 = a_1 = \sum_{j=1}^{1} a_j$. 3 pts

Maintenance: Assume that the invariant holds at the beginning of an iteration i:

$$s_k = \sum_{j=1}^{k} a_j = a_1 + ... + a_k \quad \forall k < i$$

We want to show it holds after that iteration.

At the end of iteration i, s_i gets updated to $s_{i-1} + a_i$ according to the algorithm.
But $i - 1 < i$, Therefore by the invariant $s_{i-1} = \sum_{j=1}^{i-1} a_j = a_1 + ... + a_{i-1}$.

Thus we get: $s_i = s_{i-1} + a_i = a_1 + ... + a_{i-1} + a_i = \sum_{j=1}^{i} a_j$

Since $s_1, ..., s_{i-1}$ were not changed, therefore at the end of the iteration:

$$s_k = \sum_{j=1}^{k} a_j = a_1 + ... + a_k \quad \forall k \leq i$$

4 pts

Termination: It is obvious that the loop terminates when $i = n + 1 > n$.

Upon Termination we have:

$$(i = n + 1) \wedge (\forall k < i \ s_k = \sum_{j=1}^{k} a_j = a_1 + ... + a_k)$$

$$s_k = \sum_{j=1}^{k} a_j = a_1 + ... + a_k \quad \forall k \leq n$$

Therefore $s = s_1, ..., s_n$ contains the prefix sum for $a = a_1, ..., a_n$.
The same proof applies for the in place algorithm by changing s to a. 3 pts

(e) (5%) Develop the run-time of the algorithm in terms of the input size.

The algorithm executes one assignment outside of the for loop $s_1 = a_1$.
The for loop contains $n - 1$ iterations, each of these iterations contain one assignment and one addition.
Total run time requires $2(n - 1) + 1 = 2n + 1$ operations $\in \Theta(n)$.

The in place version has an equivalent for loop, but it does not have an assignment before it.
Total run time for in place requires $2(n - 1)$ operations $\in Theta(n)$.

(f) (2.5%) Determine the amount of space required by the algorithm in terms of the input size.

In addition to the input list a, the algorithm requires space for the result list $s = s_1, ..., s_n$.
Overall, the algorithm requires space for $n + n = 2n \in \Theta(n)$ items (variables).
n for s and n for the input a.

The in place algorithm operates only on the input without requiring auxiliary space (no s).
Overall, the in place algorithm requires space for n items (variables) which are the input list $a_1, ..., a_n$.