Student ID: _____

AUT UNIVERSITY COMPUTER + MATHEMATICAL SCIENCES

716181 – Algorithm Design and Analysis

Semester 2, 2015

Mid-Semester Test (Mock)

Time Allowed 90 Minutes (plus reading time).

Instructions

- This is a closed-book exam.
- Please answer all questions in the space provided.
- Calculator is allowed for this exam.

Question	Marks given	Marks possible
1		10
2		10
3		10
4		10
5		10
6		10
Total		60

1. Answer the following questions:

   (a) Describe the main steps of Karatsuba's algorithm. (5 *marks*)

   (b) Analyse the running time of Karatsuba's algorithm using a recursive expression of the form (3 *marks*)

   $$T(n) = aT(n/b) + f(n)$$

   Then apply master's theorem on the above form and express the time complexity of the algorithm in Θ-notation. Show all working.

   (c) A program that implements the *merge sort* algorithm spends exactly 1 millisecond to sort 1,000 data items. Estimate how long this program will sort 1,000,000 items. (2 *marks*)

2. Answer the following questions:

(a) Assume that each of the expressions below gives the processing time $T(n)$ spent by an algorithm for solving a problem of size n. Order these expressions in increasing asymptotic complexity. (3 *marks*)

- $f_1(n) = n^2 + 5n(\log_2 n)^2$
- $f_2(n) = n^{1/2} \log_2 n + 1000n$
- $f_3(n) = 0.003 \log_2 \log_2 n$
- $f_4(n) = 1000$
- $f_5(n) = 5 \times 2^n + n^3$
- $f_6(n) = \frac{3^n}{10} + 5$

(b) Determine whether each statement is TRUE or FALSE. (3 *marks*)

(i) Statement: If $f(n)$ is $\Theta(g(n))$ and $g(n)$ is $O(h(n))$, then $f(n)$ is $\Theta(h(n))$.
TRUE or FALSE?

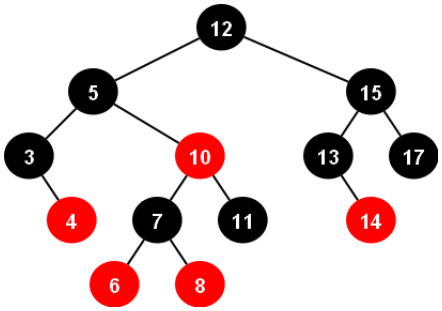
(ii) Statement: If $f(n)$ is $O(h(n))$ and $g(n)$ is $O(h(n))$, $f(n) + g(n)$ is $O(h(n))$.
TRUE or FALSE?

(iii) Statement: If $f(n)$ is $O(g(n))$ and $h(n)$ is $\Omega(g(n))$, then $f(n)h(n)$ is $\Theta(g^2(n))$.
TRUE or FALSE?

(c) Describe how you would implement the DFS algorithm on a directed graph without recursion. (4 *marks*)

3. The following diagram shows a red-black tree (with the NIL leaves invisible).

   (a) Perform the insertion algorithm to the following red-black tree to insert a node with key 9. Show all steps. (6 *marks*)



   (b) The main advantage of a red-black tree against a normal binary search tree is that the height of a red-black tree is kept low, compared to the size of the tree. (4 *marks*)

      (i) Suppose a red-black tree contains n nodes. What is the maximum height of this tree?

      (ii) What properties of a red-black tree guarantee that its height is small compared to its size? State the properties (not just their names).

4. Answer the following questions:

(a) The following is the adjacency matrix of a digraph with 6 nodes $\{0, 1, 2, 3, 4, 5\}$. (5 *marks*)

List all out-neighbours of the node 5? [1 mark]

What is the in-degree of the node 4? [1 mark]

Is the graph linearisable? [1 mark]

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Write down the adjacency list of this digraph. [1 mark]

Draw this graph. [1 mark]

(b) We define a *complete bipartite graph* $K_{s,t}$, where s, t are positive integers, as an undirected graph whose nodes are $V = \{a_1, a_2, \ldots, a_s, b_1, b_2, \ldots, b_t\}$ and whose edges are $E = \{\{a_i, b_j\} \mid 1 \leq i \leq k, 1 \leq j \leq \ell\}$. (3 *marks*)

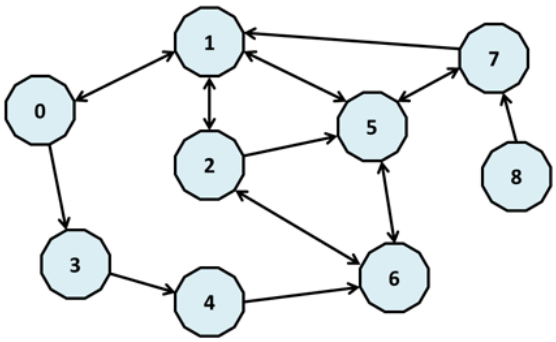
- Draw the complete bipartite graph $K_{3,3}$. [1 mark]

- What are the order and size of the graph $K_{s,t}$ [1 mark]

- Suppose we run DFS from a node of $K_{s,t}$. What is the height of the DFS forest? [1 mark]

(c) What does it mean for a directed to be *strongly connected*? (2 *marks*)

5. Perform DFS on the digraph below starting from the node 0. Assume that whenever we have more than one nodes to choose, we always pick the smallest number.



(a) Label the pre/post times for all nodes. [3 marks]

Nodes	0	1	2	3	4	5	6	7	8
Pre									
Post									

(b) Identify the forward edges, back edges and cross edges. Label forward edges with "F", back edges with "B" and cross edges with "C" beside each edge. [3 marks]

(c) Based on the answer above, what would be the outcome of the `DFS-Linearize`(G) algorithm when it is run on the above graph? [1 mark]

(d) What is the output of the Kosaraju-Sharir algorithm when it is run on the above graph? [3 marks]

6. An array $A[1..n]$ contains all the numbers from 0 to n except one. Our goal is to determine which number between 0 and n is missing in A. This time, however, we cannot access an entire integer in A with a single operation. The elements of A are represented in binary, and the only operation we can use to access them is "*fetch the jth bit of $A[i]$*", which takes constant time.

   For example, a possible input array A for $n = 7$ could be of the form

   $$[001, 110, 000, 011, 111, 101, 010]$$

   The operation $fetch(i, j)$ will return us a 0/1 value, e.g., $fetch(4, 1)$ will return 0 as the first bit of $A[4] = 011$ is 0, $fetch(6, 3)$ will return 1 as the 3rd bit of $A[6] = 101$ is 1. In this case, the missing number is $4 = 100$.

   Describe how you would design an algorithm to determine the missing number. Your algorithm should be as efficient as possible. What is the running time of your algorithm? (hint: An ideal solution to this question would run in time $O(n)$)

   (a) Informally describe your algorithm using plain English. (3 *marks*)

   (b) Write down the pseudocode of the algorithm (4 *marks*)

   (c) Analyse the time complexity of your algorithm and show that it indeed runs in time $O(\log n)$. (3 *marks*)