CSC 225 SPRING 2012
ALGORITHMS AND DATA STRUCTURES I
FINAL EXAMINATION
UNIVERSITY OF VICTORIA

1. Student ID: ____________________
2. Name: ____________________
3. DATE: 12 APRIL 2012
   DURATION: THREE HOURS
   INSTRUCTOR: V. SRINIVASAN
4. THIS QUESTION PAPER HAS 14 PAGES (INCLUDING THE COVER PAGE).
5. THIS QUESTION PAPER HAS 8 QUESTIONS.
6. ALL ANSWERS ARE TO BE WRITTEN ON THIS EXAMINATION PAPER.
7. THIS IS A CLOSED BOOK EXAM. NO AIDS ARE ALLOWED.
8. KEEP YOUR ANSWERS SHORT AND PRECISE.

Q1 (10)	
Q2 (10)	
Q3 (10)	
Q4 (10)	
Q5 (10)	
Q6 (10)	
Q7 (10)	
Q8 (10)	
TOTAL (80) =	

1. [10 Marks] For each of the following, give the correct answer by circling one of the two choices, TRUE or FALSE.

(a) The average-case running time of an algorithm, over all inputs of size n, is always more than its worst-case running time. TRUE FALSE

(b) $\log_b(a * c) = \log_b a \log_b c$. TRUE FALSE

(c) Stacks support insertions and deletions in the first-in first-out (FIFO) principle. TRUE FALSE

(d) An array of size n can support *insertAtRank* operation in worst-case $O(1)$ time. TRUE FALSE $\boxed{\text{NIS}}$

(e) A heap is an implementation of the ADT priority queue. TRUE FALSE

(f) A binary search tree with n leaves always has $n-1$ internal nodes. TRUE FALSE

(g) The worst-case running time of Insertion Sort and Merge Sort are equal in the asymptotic sense. TRUE FALSE

(h) The median of a set of n elements can be found in $O(n)$ time. TRUE FALSE $\boxed{\text{NIS}}$

(i) A complete, directed graph G on n vertices and m edges has $m = n(n-1)$ edges. TRUE FALSE

(j) The adjacency-matrix data structure for graphs can support the *areAdjacent(u,v)* operation in $O(1)$ time. TRUE FALSE

Questions not in the syllabus are not answered and are marked as NIS

2. (i)[5 Marks] Order the following functions by ascending growth rates:

n^n, 2^{2^n}, 2^{100}, $\sqrt{n}$, $2^{\log(n!)}$, $\frac{1}{n^2}$, n^3, $2^{n\log n}$, 5^n.

$\frac{1}{n^2}$, 2^{100}, $\sqrt{n}$, n^3, 5^n, $n!$, $2^{n\log n}$, n^n, 2^{2^n}

(ii)[5 Marks] Give the definition of "Little-Oh". Using the definition, show that $n = o(n \log n)$.

$\boxed{\text{NIS}}$

3. (i)[4 Marks] What are the three main operations supported by Dictionary ADT? What the worst-case running time of these operations when a dictionary ADT for n key-element items is implemented as a binary search tree? What is an example that shows the worst-case running time?

Three operations are : Insert (x)

Delete (x)

Find (x)

The worst-case running times of all the three operations are O(n)

Example: Insert (1), Insert(2) ...., Insert (n)
Followed by Delete (n) and Find (n-1)
The last three op. take O(n) time.

(ii)[6 Marks] Draw the AVL Tree resulting from the insertion of the key 46 into the AVL Tree shown below. Show all the steps of the restructuring procedure, if any.

NIS

4. (i)[7 Marks] Explain why any comparison-based sorting algorithm has a lower bound of $\Omega(n \log n)$ for its worst-case running time.

$\boxed{\text{NIS}}$

(ii)[3 Marks] What is the running time of Bucket-Sort on a sequence S of n items whose keys are integers in the range $[0, 3n]$? Why does this running time not contradict the lower bound of Part (i)?

- The running time of Bucket-Sort is $O(n+N)$ where n = # of items and N is the range / (ie) items are from $[0, N]$
- Since $N = 3n$,
  running time $= O(n + 3n) = O(n)$
- This does not contradict the $\Omega(n \log n)$ lower bound as it only works with the conditions that items come from $[0, N]$

5. (i) [5 Marks] Solve the following recurrence equation using repeated substitution to get a closed-formula for $T(n)$.

$$
\begin{aligned}
T(n) &= 1 \text{ if } n = 1 \\
&= 4T\left(\frac{n}{2}\right) + n \text{ if } n \geq 2
\end{aligned}
$$

$$
\begin{aligned}
T(n) &= 4T\left(\frac{n}{2}\right) + n \\
&= 4\left[4T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n \\
&= 4^2 T\left(\frac{n}{4}\right) + 2n + n \\
&= 4^2\left[4T\left(\frac{n}{8}\right) + \frac{n}{4}\right] + 2n + n \\
&= 4^3 T\left(\frac{n}{8}\right) + 4n + 2n + n
\end{aligned}
$$

<u>After i Steps</u>

$$T(n) = 4^i T\left(\frac{n}{2^i}\right) + \sum_{j=0}^{i-1} 2^j n$$

<u>Substitute $i = \log_2 n$</u>

$$
\begin{aligned}
T(n) &= 4^{\log_2 n} \underbrace{T(1)}_{= 1} + n \sum_{j=0}^{\log_2 n - 1} 2^j \\
&= 2^{2\log_2 n} + \sum_{j=0}^{\log_2 n - 1} 2^j \\
&= n^2 + n\left(\frac{2^{(\log_2 n - 1)+1} - 1}{2 - 1}\right) \\
&= \underline{n^2 + n(n-1)} = \underline{\underline{2n^2 - n}}
\end{aligned}
$$

(ii)[5 Marks] Solve the recurrence equation in Part (i) using the Master Theorem given below. Check that the answer is consistent with Part (i).

$$\begin{aligned} T(n) &= c \text{ if } n < d \\ &= aT\left(\frac{n}{b}\right) + f(n) \text{ if } n \geq d \end{aligned}$$

(a) If there is a small constant $\epsilon > 0$ such that $f(n)$ is $O(n^{\log_b a - \epsilon})$, then $T(n)$ is $\Theta(n^{\log_b a})$.

(b) If there is a small constant $k \geq 0$ such that $f(n)$ is $O(n^{\log_b a} \log^k n)$, then $T(n)$ is $\Theta(n^{\log_b a} \log^{k+1} n)$.

(c) If there are small constants $\epsilon > 0$ and $\delta < 1$ such that $f(n)$ is $\Omega(n^{\log_b a + \epsilon})$ and $af(n/b) \leq \delta f(n)$, then $T(n)$ is $\Theta(f(n))$.

$$\begin{aligned} T(n) &= 1 \text{ if } n < 2 \\ &= 4T\left(\frac{n}{2}\right) + n \text{ if } n \geq 2 \end{aligned}$$

$\therefore a = 4, b = 2, f(n) = n$

Note: $n^{\log_b a} = n^{\log_2 4} = n^2$

$\therefore f(n) = n = O(n^{2-\varepsilon})$ for $\varepsilon = 0.1$

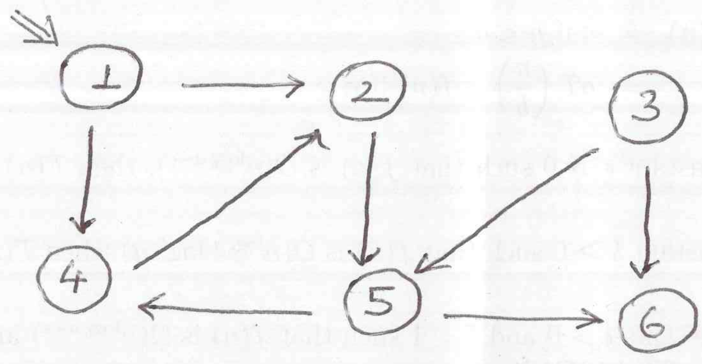
as $n = O(n^{1.9})$

$\therefore$ Case (a) applies to this reccurrence equation

$T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$

$$\boxed{T(n) = \Theta(n^2)}$$

6. Consider the following directed graph G:



(i)[5 Marks] Perform a DFS and BFS traversal on the graph shown above starting at vertex 1 and list the vertices in the order in which they are visited. Assume that, in the traversal, the adjacent vertices of a given vertex are visited in the increasing order of the vertex labels.

DFS

Preorder: 1 2 5 4 6 3

Postorder: 4 6 5 2 1 3

BFS

~~1 2 4 5 4 6 3~~

1 2 4 5 6 3

(ii)[5 Marks] A graph is said to be two-colourable if the vertices can be assigned one of two colours in such a way that no edge connects vertices of the same colour. How can you use DFS to check if a given graph is two-colourable?

- To check, we will use two Colors, Red and Blue.
- Assign an arbitrary color to the start vertex, say Red
- During the DFS traversal, when a vertex is visited (for the first time), assign it a color that is different from the vertex ~~from the vertex~~ that is its parent in the DFS tree.
- Look for a back edge that connects two vertices of same color. If so, G is not two-colorable. otherwise, it is.

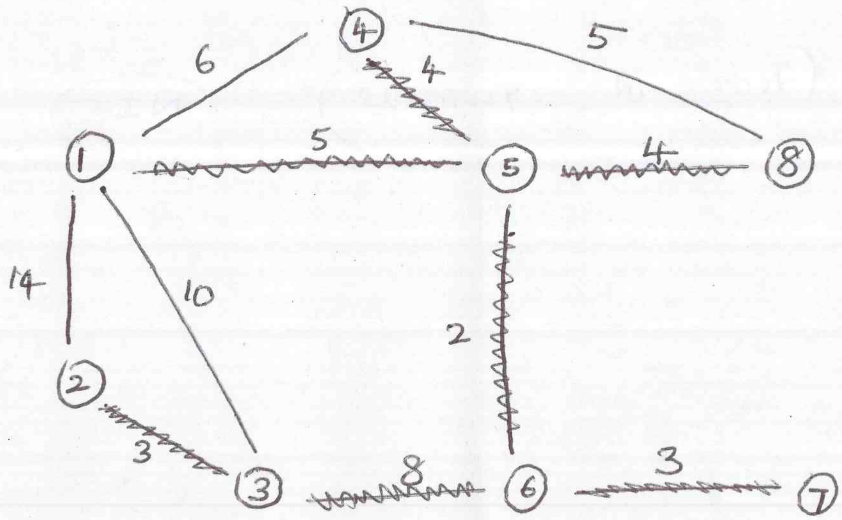
7. (i) [6 Marks] Given a weighted Directed Acyclic Graph (DAG) G on n vertices and m edges, describe in pseudo-code, a $O(n + m)$-running time algorithm that computes the shortest path from a source vertex s to every other vertex in G. You can assume that you are given the subroutine for topological sorting.

NIS

(ii) [4 Marks] How will you modify your algorithm if you want to compute the longest path from the source vertex to every other vertex in G instead of the shortest path.

NIS

8. Consider the following undirected, weighted graph G:



(i)[5 Marks] Find the minimum spanning tree of G using ~~Prim's~~ Kruskal's algorithm starting at vertex 1. Clearly, list all the edges in the order in which they are discovered. What is the weight of the minimum spanning tree?

Kruskal's algorithm sorts edges by weight and adds edges one by one to the MST as long as it does not create a cycle with previously added edges.

<u>edges added are :</u>

- (5, 6) 2
- (2, 3) 3
- (6, 7) 3
- (4, 5) 4
- (5, 8) 4
- (1, 5) 5
- (3, 6) 8

Total weight = 29

(ii)[5 Marks] Give a proof of correctness that explains why Prim's algorithm always outputs the minimum spanning tree (Hint: Prove and use the cut property).

(NOT IN SYLLABUS) NIS

END OF EXAM