CSC 225 A01 FALL 2015 (CRN: 10789)
ALGORITHMS AND DATA STRUCTURES I
FINAL EXAMINATION
UNIVERSITY OF VICTORIA

1. Student ID: ______________________
2. Name: ______________________
3. DATE: 21 DECEMBER 2015
   DURATION: 3 HOURS
   INSTRUCTOR: V. SRINIVASAN
4. THIS QUESTION PAPER HAS 13 PAGES INCLUDING THE COVER PAGE.
5. THIS QUESTION PAPER HAS TEN QUESTIONS.
6. ALL ANSWERS TO BE WRITTEN ON THIS EXAMINATION PAPER. USE THE LAST PAGE FOR CALCULATIONS.
7. THIS IS A CLOSED BOOK, CLOSED NOTES EXAM. NO CALCULATORS ARE ALLOWED.
8. **READ THROUGH ALL THE QUESTIONS AND ANSWER THE EASY QUESTIONS FIRST**. KEEP YOUR ANSWERS SHORT AND PRECISE.

Q1 (5)	
Q2 (5)	
Q3 (5)	
Q4 (5)	
Q5 (5)	
Q6 (5)	
Q7 (5)	
Q8 (5)	
Q9 (5)	
Q10 (5)	
TOTAL (50) =	

1. Compute the order of growth (as a function of N) of the running times of each of the following code fragments: [5 Marks]

(a) **Algorithm** Loop1(N)

```
int s = 0;
for (int i=1; i < N; i++)
   s = s++;
```

(b) **Algorithm** Loop2(N)

```
int s = 0;
for (int i=1; i < N; i++)
   for (int j=0; j < N; j++)
      s = s++;
```

(c) **Algorithm** Loop3(N)

```
int s = 0;
for (int i=1; i < N; i *= 2)
   s = s++;
```

(d) **Algorithm** Loop4(N)

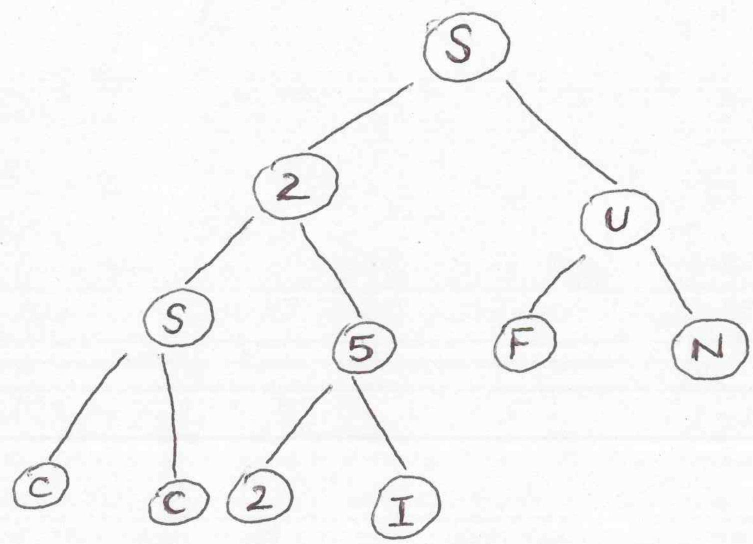
```
int s = 0;
for (int i=1; i < N; i++)
   for (int j=0; j < i; j++)
      s = s++;
```

(e) **Algorithm** Loop5(N)

```
int s = 0;
for (int i=1; i < N; i++)
    for (int j=0; j < i; j *=2)
        s = s++;
```

2. **Basic Data Structures**

(a) For the binary tree below, output the sequence of nodes in the order in which they are visited in the preorder, postorder and inorder traversal. [3 Marks]



Traversal	Sequence
Preorder	
Postorder	
Inorder	

(b) Describe the steps of an algorithm (in clear pseudo-code) that constructs and outputs a binary tree, given its preorder and inorder traversal sequences as input. You need not compute its running time. [2 Marks]

3. (a) For each of the sorting algorithms given in the table below, write down its best case and worst case running times. [3 Marks]

Algorithm	Best-Case	Worst-Case
Insertion-Sort		
Merge-Sort		
Quick-Sort		

(b) You are given a set of n distinct keys. After a preprocessing algorithm you should be able to compute queries of the following form in $\mathbf{O(1)}$ time:
Input: Two integers i and j, $1 \leq i \leq j \leq n$
Output: The sum of the ith largest, plus the $(i + 1)$st largest, plus the $(i + 2)$nd largest, up until the jth largest.

Describe the steps of the preprocessing algorithm that you need. What is the total running time of your preprocessing algorithm? [2 Marks]

4. **[The Master Theorem]**

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

(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 $\Theta(f(n))$.

Solve the following recurrence equations using the Master Theorem given above (assuming that $T(n) = c$ for $n < d$, for constants $c > 0$ and $d \geq 1$). [5 Marks]

(a) $T(n) = 16T(n/2) + n^4 \log n$

(b) $T(n) = 16T(n/2) + n^5$

(c) $T(n) = 4T(n/2) + n$

5. (a) Insert items with the following keys (in the given order) into an initially empty binary search tree:

$$30, 40, 24, 58, 48, 26, 11, 13$$

Draw the tree after each insertion. [3 Marks]

(b) Delete the item with key 24 from the binary search tree obtained at the end of all the insertions in Part (i). Draw the tree after the deletion. [1 Mark]

(c) For each of the following operations supported by a binary search tree, write down its worst-case running time for an input of size n. [1 Mark]

Operation	Worst-Case
findItem	
deleteItem	
insertItem	

6. **Hashing**

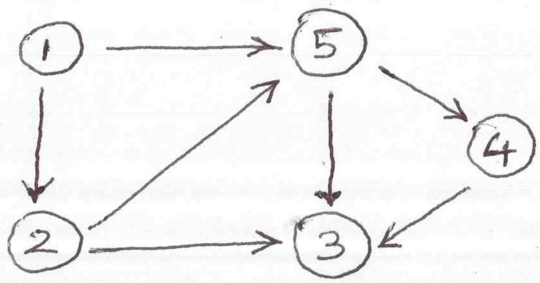
(a) Suppose that a hashing scheme stores a set of 10 keys

5 11 19 15 20 33 12 17

using the hash function $h(k) = (2k+4) \bmod 9$. It uses open addressing and resolves collisions using linear probing. Draw the hash table with all the 10 keys inserted. [4 Marks]

(b) What is the load factor of the table? [1 Mark]

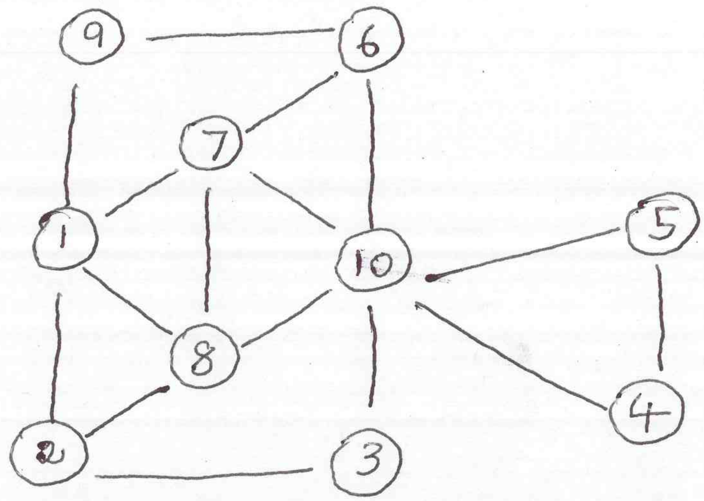
7. Consider the following directed graph G.



(a) Give the adjacency list and adjacency matrix representations of G. [3 Marks]

(b) Compute the adjacency matrix of $G^2 = G \times G$. G^2 is a graph that has the same set of vertices as G and contains an edge between two vertices if there is a path of length 2 between them in G. [2 Marks]

8. Consider the following undirected graph G.

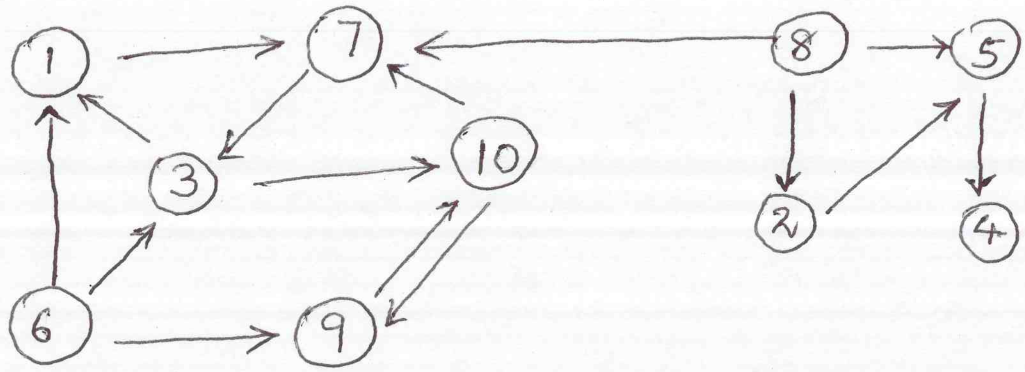


(a) Perform a BFS traversal on G starting at vertex 1. Assume that, in the traversal, the adjacent vertices are visited in the increasing order of vertex labels. List the BFS tree edges in the same order in which they are found. [3 Marks]

(b) Output the first edge during the BFS traversal that indicates the presence of an even cycle in G. [1 Mark]

(c) Output the first edge during the BFS traversal that indicates that G is not bipartite. [1 Mark]

9. Consider the following directed graph G.



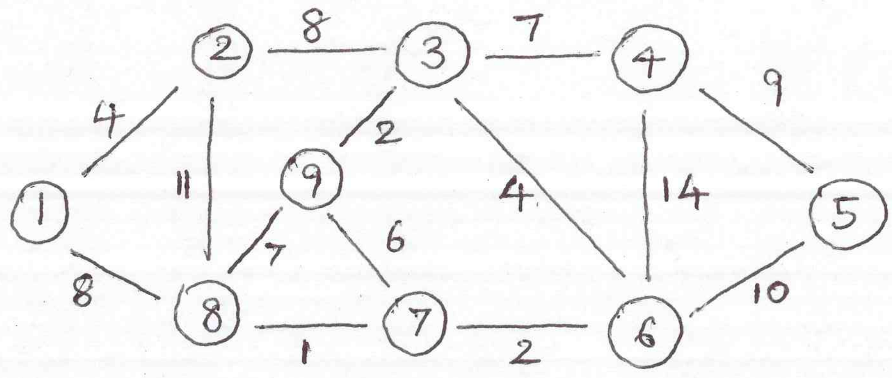
(a) Perform a DFS traversal on G starting at vertex 1. Assume that, in the traversal, the adjacent vertices are visited in the increasing order of vertex labels. Compute the preorder and postorder listing of the vertices. [3 Marks]

Traversal	Sequence
Preorder	
Postorder	

(b) Using Kosaraju's algorithm, compute the strongly connected components of G. [2 Marks]

10. **Minimum Spanning Trees.**

Consider the following undirected, weighted graph G.



(a) Find the minimum spanning tree (MST) of G. Clearly list all the edges in the order in which they are discovered by your algorithm. What is the weight of the MST? [4 Marks]

Edge Label	Weight
Total Weight	

In the table above, an edge label is the pair of end points of that edge.

(b) Suppose that you are asked to compute the *Maximum Weight Spanning Tree*, the spanning tree with the maximum weight. However, you are not allowed to modify Kruskal's or Prim's algorithm in any way. How will you modify the input graph so that you can compute the maximum weight spanning tree using Kruskal's or Prim's algorithm. [1 Mark]

BLANK PAGE FOR CALCULATIONS

END OF EXAM