Index No	

UNIVERSITY OF MORATUWA
Faculty of Engineering

BSc Engineering Level 2, 2003/2004
Semester 1 Examination (Held in January 2004)

CS222 Algorithms

Answer All Questions **Time: 2 Hours**

Note: *Answers must be given in the space provided. Write your Index No. top of this page.*

1. Let A be a sorted array of n=10 elements. We are using a binary search to find an element k. Assume that k is in the array. Also assume that only one comparison is required to determine whether k is equal to, less than or greater than A[i]. What is the average number of comparisons required if k can be in any arbitrary location? (Hint: If k is in the middle of the array, then only one comparison is required). [10 marks]

……

……

……

……

……

……

……

2. Consider the following recursive algorithm, COMP, where n is a non-negative integer.

```
COMP(n)
    if n = 0
            return 0
    else
            return (2*n + COMP(n-1))
```

What value will be returned if we invoke this with n=6? [5 marks]

……

……

……

……

……

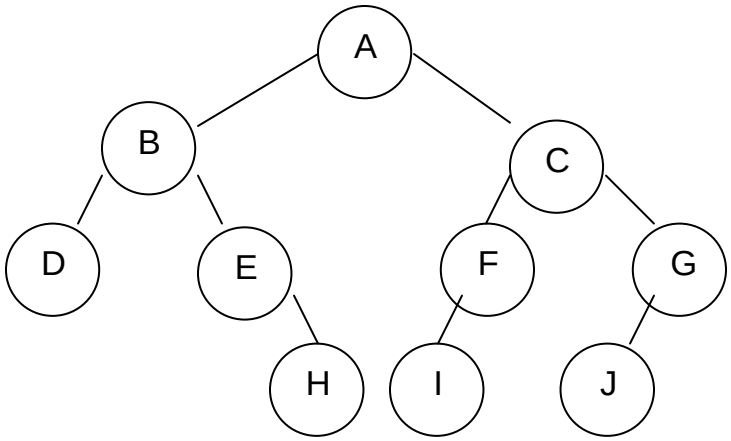
3. Give the order of nodes corresponding to *postorder* traversal of the binary tree in the figure shown. [5 marks]

……..

……..

……..

……..



4. For the following running times for algorithms, show the correct relationship among them using the "<" (less than) relation: $O(n)$, $O(\lg n)$, $O(n \lg n)$, $O(2^n)$, $O(n^2)$. [5 marks]

……

……

……

5. Suppose you have been given the following algorithm which accepts an array A[1..*n*] of *n* integers as input.

```
for i ← n downto 2
        large ← A[1]
        index ← 1
        for j ← 2 to i
                if A[j] > large
                        large ← A[j]
                        index ← j
        A[index] ← A[i]
        A[i] ← large
```

Compute the running time of this algorithm and express it in "big oh" notation. [6 marks]

……

……

……

……

……

……

……

……

6. What is the purpose of the algorithm given in question 5? (i.e., what is it doing?) [4 marks]

……

……

……

……

7. Consider the following *fractional-knapsack problem*. Suppose you are given 5 objects with the following weights:

(6, 10, 9, 5, 12)

Suppose the following profits are associated with the objects (in the same order):

(8, 5, 10, 15, 7)

The knapsack has a weight capacity of 30. The objective is to fill the knapsack while maximizing the total profit. Note that a fraction of an object can be chosen if desired but you may do it only once for one object. Compute the maximum profit possible. [10 marks]

……

……

……

……

……

……

……

……

……

……

……

……

8. Give the asymptotic growth in "*big oh*" notation for the following functions of $n > 0$. [4 marks]

$8n^9 + n^{2n} + 5n^3 + 3n + 9$ ……..

$5\log n + 4^n + 2n + 18$ ……..

$15n^5 + (3n + 7)^6$ ……..

$9999n + 0.003\, n^2 + 0.9^n$ ……..

9. The following algorithm is to find the maximum value in a given integer array, A[1..*n*], of length *n*.

```
max ← A[1]
for i ← 2 to n
        // missing part
return max
```

Give the missing part of this algorithm. [4 marks]

……

……

……

10. Give a recursive algorithm corresponding to the following expression to compute *Ack*(*m*, *n*) for any two given integers $m, n \geq 0$. [6 marks]

$$Ack(m,n) = \begin{cases} n+1 & if \quad m = 0 \\ Ack(m-1,1) & if \quad m \neq 0, n = 0 \\ Ack(m-1, Ack(m,n-1)) & otherwise \end{cases}$$

……

……

……

……

……

……

……

……

……

……

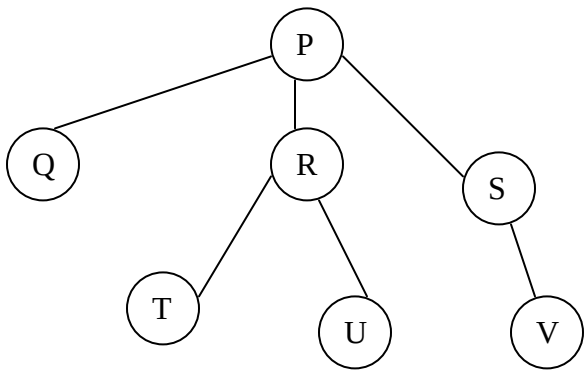
11. Show the order of vertices visited in the following graph if the graph is traversed
a) depth-first
b) breadth-first
starting at vertex S. [6 marks]

a) …….

…….

b) …….

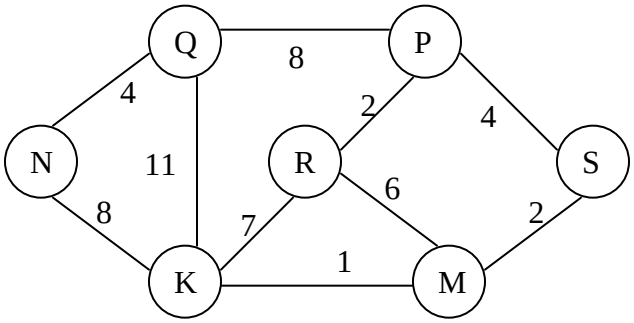
……..



12. *Kruksal's algorithm* for computing the minimum-cost-spanning tree (MCST), *T*, for a graph, *G*=(*V*, *E*), is given below.

```
MCST-Kruksal(G, W)                    // G=(V, E), W is the array of weights
    T ← Ø
    for each vertex v ∈ V
        Make-Set(v)                   // make separate sets for vertices
    Sort the edges by increasing weight
    for each edge (u,v) ∈ E, in sorted order
        if Find-Set(u) ≠ Find-Set(v)  // if no cycles are formed
            T ← T ⋃ {(u,v)}           // add edge to tree
            Union(u,v)                // combine sets
    return T
```

Using this (or other) algorithm, find the MCST for the following graph. [10 marks]



……

……

……

……

……

……

……

……

……

……

13. Suppose you have a 100,000-character data file that you wish to store compactly using a variable-length code. There are 6 different characters that occur. Their frequencies of occurrences are as follows.

Character	*K*	*m*	*n*	*p*	*q*	*s*
Frequency (%)	44	13	11	17	9	6

a) Show the variable-length codeword for each character obtained using an appropriate algorithm (e.g., Huffman's algorithm). [8 marks]

b) What is total number of bits needed to represent the original data file using this code? [2 marks]

……

……

……

……

14. Suppose *S* is a stack data structure with the contents {10, 4, 5, 7, 2} where the top element is 10. Then suppose the following code segment is executed on it.

```
x ← POP (S)
print x
x ← POP (S)
print x
PUSH (S, 3)
PUSH (S, 6)
x ← POP (S)
print x
x ← POP (S)
print x
```

After executing the code segment, show
a) the printed result, and [5 marks]
b) the contents of the stack. [5 marks]

a)……

……

……

……

……

b)……

……

……

……

……

15. A student, X, tries to prove that a problem Y is **NP**-complete using the following steps, in that order.

- **(i)** X shows that Y is in **NP**
- **(ii)** X selects a known **NP**-complete problem Z
- **(iii)** X shows how Y can be polynomially reducible to Z
- **(iv)** X therefore concludes Y is **NP**-complete

Is this approach correct or incorrect? Explain your answer. [5 marks]

……

……

……

……

……

……

……

……