Sri Lanka Institute of Information Technology

B. Sc. Special Honours Degree/ Diploma

In

Information Technology

Final Examination

Year 2, Semester 2 (2017)

IT240 – Design and Analysis of Algorithms

Duration: 3 Hours

Instructions to Candidates

- This paper contains 6 questions on 8 pages.
- Each question carries equal marks.
- Answer ALL questions.
- Read all questions before start answering.
- The total marks obtainable for this examination is 120.
- This examination accounts for 70% of the module assessment.
- This is a close book examination.

Question One (20 marks)

a) Briefly describe the following terms as related to tree terminology. (2 marks)
   i. Binary Tree
   ii. Leaf node

b) Prove the relationship between nodes (n) and height (h) of Full Binary Tree is given by the formula $2^h = (n+1)/2$.

(4 marks)

c) Compare and contrast the priority queue and the first in first out queue. (2 marks)

d) Consider the following min-heap A = <1, 8, 10, 15, 11, 12, 16, 40, 50>. Assume that the key 10 in the min-heap is modified to 20. Create the resulting max-heap after the modification.

(4 marks)

e) We can compute the upper bound on the running time of BUILD-HEAP as follows.

$$T(n) = \sum_{h=o}^{\lfloor \lg n \rfloor} \lceil a \rceil O(b)$$

Find the expressions for a and b. (2 marks)

f) Illustrate the operation on the following heap for HEAP_INSERT(A,100). (For the purpose of illustration, assign the values only once to the given algorithm and use diagrammatic way to reach the answer.) (6 marks)

1	2	3	4	5	6	7	8
80	30	50	20	12	45	9	0

```
Procedure HEAP_INSERT(A, key)

    heap_size[A] ← heap_size[A]+1
    i ← heap_size[A]
    while i > 1 and A[PARENT(i)] < key
          A[i] ← A[PARENT(i)]
          i ← PARENT(i)
    A[i] ← key
```

QUESTION TWO (20 marks)

(a) Briefly describe the following terms in relation to the Greedy Method. (2 marks)

i. Disjoint Data Structure
ii. FIND-SET operation

(b) The algorithm given below is to find the Minimum Cost Spanning Tree (MCST) of a graph.

KRUSKAL'S ALGORITHM

```
Procedure MST-KRUSKAL(G, w)
A ← Ø
for each vertex v∈ V[G]
      do MAKE-SET(v)
sort the edges of E into no decreasing order by weight w
for each edge (u,v) ∈ E, taken in no decreasing order by weight
      do if FIND-SET(u)≠ FIND-SET(v)
            then A ← AU{(u, v)}
                  UNION(u, v)
return A
```

i. Find the Minimum Cost Spanning Tree (MCST) of the graph given below using Kruskal's algorithm and calculate the minimum cost. (For the purpose of illustration, assign the values only once to the given algorithm and use diagrammatic way to reach the answer.) (9 marks)

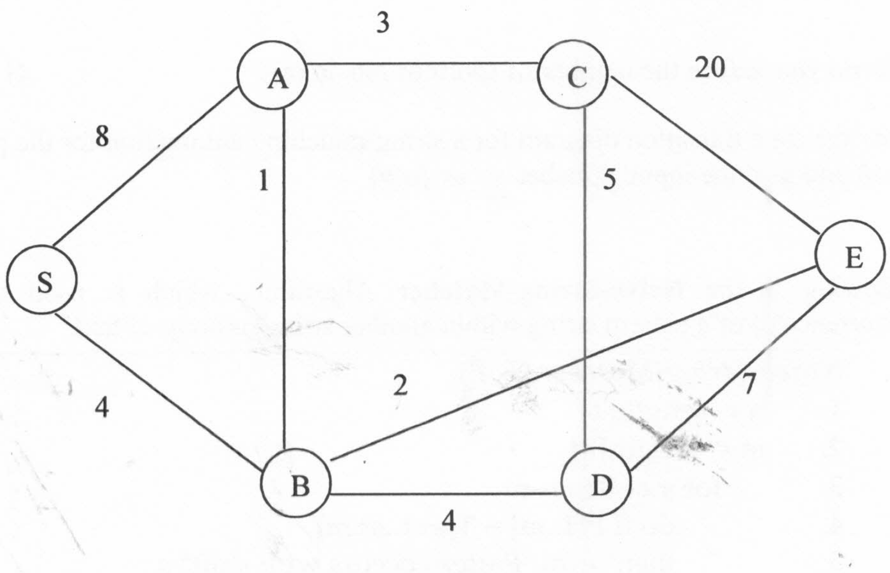


Fig. 01

ii. Find the Minimum Cost Spanning Tree (MCST) of the graph given in Fig. 01 using Prim's algorithm and calculate the minimum cost. (For the purpose of illustration, assign the values only once to the given algorithm and use diagrammatic way to reach the answer.) Starting node is S. (9 marks)

PRIM'S ALGORITHM

```
MST-PRIM(G, w, r)
for each u ∈ V[G]
        do key[u] ← ∞
                π[u] ← NIL
key[r] ← 0
Q ← V[G]
while Q ≠ Ø
        do u ← EXTRACT-MIN(Q)
                for each v ∈ Adj[u]
                do if v ∈ Q and w(u, v) < key[v]
                        then π[v] ← u
                        key[v] ← w(u, v)
```

Question Three (20 Marks)

a) Taking modulo $q = 9$, how many spurious hits and valid hits do the Rabin -Karp matcher encounter in the text T = 40203010602010 when looking for pattern P =20? (4 marks)

b) How do you reduce the number of spurious hits in (a)? (1 marks)

c) Draw the state transition diagram for a string-matching automation for the pattern P = ***aaab*** and take the input alphabet Σ as $\{a,b\}$ (10 marks)

d) Following is the **Naïve-String-Matcher** Algorithm, which is used to find the occurrence(s) of a pattern string within another string or body of text.

```
Naïve-String-Matcher (T, P)
    n ← length[T]
    m ← length[P]
        for s ← 0 to n-m
            do if P[1..m] = T[s+1..s+m]
            then print "Pattern occurs with shift" s
```

Given the text and pattern as follows;

Text ***T***

1	0	0	0	1	0	0	0	1

Pattern ***P***

1	0	0

i. How many comparisons would occur in this algorithm?

ii. Show that best-case time complexity of the above algorithm is $O((n-m+1))$ where n is the number of characters in the text and m is the number of characters in the pattern.

(5 marks)

QUESTION FOUR (20 marks)

(a) What do we need code like ASCII and Unicode in data representation? (2 marks)

(b) A file has alphabet {A, B, C, D, E), where A has frequency 35, B has frequency 10, C has frequency 20, D has frequency 5, and E has frequency 80.

i. Develop a code word for the above symbols using Shannon Fano Algorithm given below. (7 marks)

Shannon Fano Algorithm.

1. *Line up the symbols by falling probability (or frequency) of incidence.*
2. *Divide the symbols into two groups, so that both groups have equal or almost equal sum of the probabilities.*
3. *Assign value 0 to the first group, and value 1 to the second.*
4. *For each of the both groups go to step 2.*

ii. Construct the codeword for the above data using Huffman Algorithm given below. (7 marks)

```
Huffman (C)
    n ← |C|
    Q ← C
    for i ← 1 to n-1
        do allocate a new node z
            left[z] ← x ← EXTRACT-MIN(Q)
            right[z] ← y ← EXTRACT-MIN(Q)
            f[z] ← f[x] + f[y]
            INSERT(Q,z)
    return EXTRACT-MIN(Q)
```

(c) Using the "**Ziv –Lempel**" model transfer the following message into the symbol stream. (4 marks)

"aabcabcaaaabcabcaa"

Letter	ASCII Value
a	01000001
b	01000010
c	01000011

QUESTION FIVE **(20 marks)**

a) Given a chain $(A_1, A_2, ..., A_{n-1}, A_n)$ of n matrices, where for $i = 1,2, \ldots, n$ matrix A_i has dimension $p_{i-1} \times p_i$. Assume that $m[i,j]$ is the minimum number of scalar multiplications needed to compute the matrix $A_{i...j} = A_i \times A_{i+1} \times ..... \times A_{j-1} \times A_j$ and it is defined below

$$m[i,j] = \begin{cases} 0 & \text{if } i = j \\ \min\limits_{i \le k < j} \{m[i,k] + m[k+1,j]\} + p_{i-1}p_k p_j & \text{otherwise} \end{cases}$$

Consider the following set of metrics A_1, A_2, A_3 and A_4 with their dimensions of 2×3, 3×5, 5×10 and 10×2 respectively.

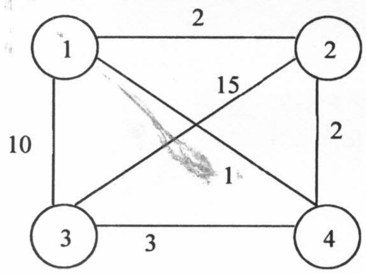
i. Draw the *m* and *s* table to find the optimal parenthesizing of the matrices for the above sequence of matrices using the dynamic programming algorithm.

ii. Hence find the optimal parenthesizing and optimal number of scalar multiplications of the above matrices

```
MATRIX-CHAIN-ORDER (p)
n ← length[p] − 1
for i ← 1 to n
     do m[i, i] ← 0
for l ← 2 to n                 ▷ l is the chain length.
     do for i ← 1 to n − l + 1
            do j ← i + l − 1
               m[i, j] ← ∞
               for k ← i to j − 1
                   do q ← m[i, k] + m[k + 1, j] + p_{i−1} p_k p_j
                      if q < m[i, j]
                        then m[i, j] ← q
                             s[i, j] ← k
return m and s
```

Question Six **(20 Marks)**

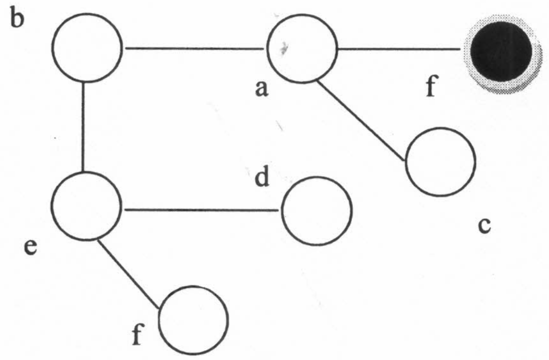
a) Draw the solution space tree for the traveling salesman problem shown in the graph given below taking node1 as the start node. (4 marks)



i. Trace the working of a backtracking algorithm on this tree. Clearly label the nodes in the order in which the backtracking algorithm first reaches them. Identify the nodes that do not get reached. (6 marks)

b) Illustrate the operation of the DFS (Depth First Search) on the graph given below taking the starting vertex as vertex *f*. (For the purpose of illustration, assign the values only once to the given algorithms at the end of the paper and use diagrammatic way to reach the answer.)

(10 marks)



```
Procedure DFS(G)
for each vertex u ∈V [G] ⊳ V [G] = {a,b,c,d,e,f}
            do color[f] ← WHITE
                   π[f] ← NIL
time ← 0
for each vertex u ∈V [G] ⊳ V [G] = {a,b,c,d,e,f}
            do if color[f] = WHITE
                    then DFS-VISIT(f)
```

```
Procedure DFS-VISIT(f)
color[f] ← GRAY
time ← 0 +1 = 1
d[u] ← 1
for each v ∈Adj[f]     ⊳ Adj[f] = {e}
            do if color[e] = WHITE
                   then π[e] ← f
```