FIRST TERM EXAMINATION

FIFTH SEMESTER [B.Tech.], September, 2016
Algorithm Design and Analysis (ETCS-301)

Time : 1.30 hours *Maximum Marks : 30*

Note: 1. Attempt three questions in total

2. Q. No. 1 is compulsory. Attempt any two more question from the remaining.

Question 1.

(a) Distinguish between O(Big Oh) and o(little Oh) notations.

(b) Define subtract and conquer master theorem.

(c) Prove that $(n+a)^b = O(n^b)$.

(d) Explain Overlappin subproblems.

(e) What Overlapping subproblems.

Solution:

(*a*) A little-o bound is a stronger condition than a big-O bound.

Big-O is an upper bound. $f(x)$ is $O(g(x))$ if $|f(x)|<cg(x)$ for some constant c and sufficiently large x.

little-o is an asymptotic limit. $f(x)$ is $o(g(x))$ if

$$\lim_{x\to\infty}\frac{f(x)}{g(x)}=0$$

Informally you can think of big-O as "grows no faster than $g(x)$" and little-o as "grows much slower than $g(x)$".

(*b*) Let's talk about Subtract and Conquer Master Theorem. Suppose we have a recurrence relation of the form

$$T(n) = aT(n-b) + f(n) \text{ when } n>1$$
$$= c \text{ when } n<=1$$

Here $f(n) = O(n^d)$ Therefore, $T(n) = aT(n-b) + O(n^d)$ when $n>1$

Now to find order of complexity for this kind of recurrence relation we can use subtract and conquer Master theorem. It states as follows:

$$T(n) = 0(n^d) \text{ if } a<1$$
$$= 0(n^{d+1}) \text{ if } a=1$$
$$= 0(n^d a^n/b) \text{ if } a>1$$

This is very helpful for finding order of complexity for recurrence relation of the form $T(n) = aT(n-b)+f(n)$ Generally we come across Master Theorem but that is not helpful for this kind of recurrence.

(*c*) Refer to Question 1(d) of First Term 2011.

(*d*) a problem is said to have overlapping subproblems if the problem can be broken down into subproblems which are reused several times or a recursive algorithm for the problem solves the same subproblem over and over rather than always generating new subproblems.

(*e*) Refer to Question 5(a) of End Term 2015.

Question 2.

Solve the following recurrence relations :

(a) $T(n)=T(n/3) + T(2n/3) + n$ *(using recursion tree)*

(b) $T(n)=4T(n/2) + n3$ *(using master method)*

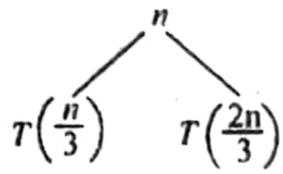
(c) $T(n)=2T(Ln/2.1) +n$ *(using substitution method)*

(d) $T(n)=\begin{cases} 1 & \text{if } n=1 \\ 2T(n/2)+n & \text{if } n>1 \end{cases}$ *(using iteration mehtod)*

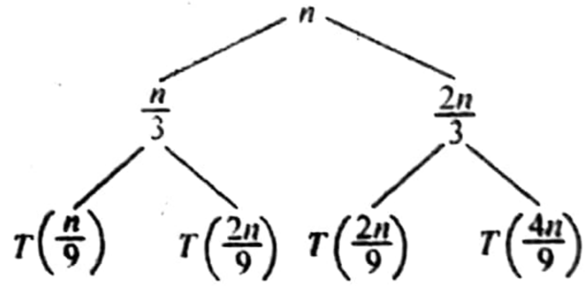
Solution:

(a) $T(n) = T\left(\frac{n}{3}\right)+T\left(\frac{2n}{3}\right)+n$

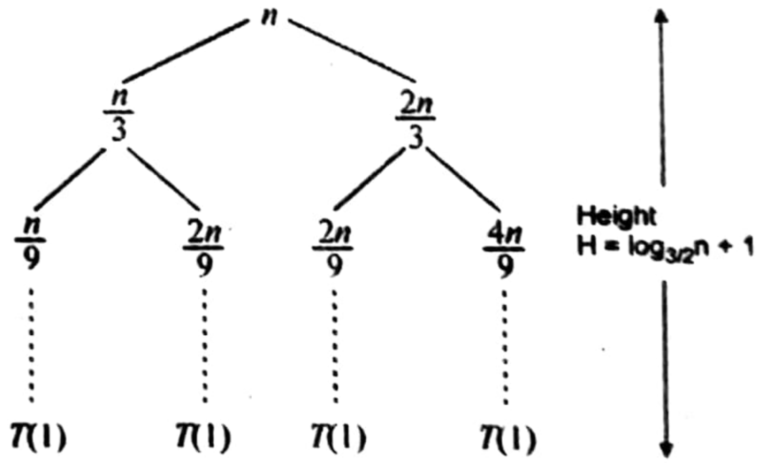
I.



II.



III.



Total complexity -

$$T(n) = \text{Cost at each level} * \text{Height}$$

$$= n*\left(\log_{\frac{3}{2}} n+1\right)$$

$$= O(n \log n)$$

(b) $$T(n) = 4T\left(\frac{n}{2}\right)+n^3$$

$a = 4, b = 2, f(n) = n^3$

$$n^{\log_b a} = n^{\log_2 4} = n^{\log_2 (2)^2} = n^2$$

Hence $$f(n) = n^{\log_b a + \varepsilon}$$

which satisfies case III of master method, and it also satisfy regularity condition-

$$a f\left(\frac{n}{b}\right) = 4 f\left(\frac{n}{2}\right) = 4\frac{n^3}{8} = \frac{1}{2}n^3$$

$$\leq C f(n) \text{ for all } C \geq \frac{1}{2}$$

Hence

$$T(n) = O(n^3)$$

(c) $T(n) = 2T\left(\frac{n}{2}\right) + n$

Step I : Assume a solution

$$T(n) = O(n \log n)$$
$$T(n) \leq C \cdot n \log n \text{ for all } n \geq n_0$$

Step II : $$T(n) \leq 2 \cdot C\left[\frac{n}{2}\right]\log\left[\frac{n}{2}\right] + n$$

As 'n' is very large, hence

$$\frac{n}{2} = \left[\frac{n}{2}\right]$$

$\therefore$ $$\leq 2 \cdot C\frac{n}{2}\log\frac{n}{2} + n$$

$$\leq C n \log\frac{n}{2} + n$$
$$\leq C n(\log n - 1) + n$$
$$\leq C n \log n - C n + n$$

Hence for $C = 1$

$$T(n) = O(n \log n)$$

(d) $$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 2T\left(\frac{n}{2}\right)+n & \text{if } n>1 \end{cases}$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$= 2\left[2T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n = 4T\left(\frac{n}{4}\right) + \frac{n}{2} + n$$

$$= 2^i T\left(\frac{n}{2^i}\right) + \frac{n}{2^{i-1}} + \ldots + n$$

for $$\frac{n}{2^i} = 1 \Rightarrow i = \log_2 n$$

$$T(n) = 2^{\log_2 n} T(1) + \frac{n}{2^{\log_2 n-1}} + \ldots + n$$

$$= n^{\log_2 2} + \sum_{k=0}^{\log_2 n-1} \frac{n}{2^k}$$

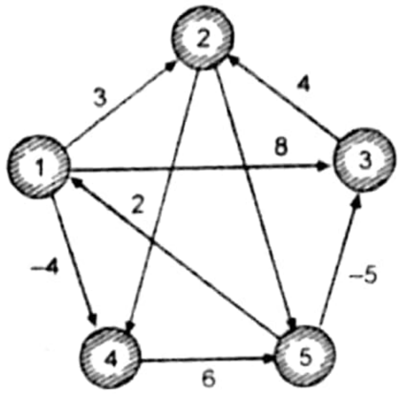
$$= O(n \log n).$$

ition-

Question 3.

(a) ***Write Insertion sort algorithm. Explain best case and worst case time complexity of Insertion sort algorithm***

(b) ***Find all pairs shortest path for the following graph using Floyd Warshall Algorithm.***



Solution:

(*a*) INSERTION-SORT (A)

```
for j <- 2 to length[A]
        do key <- A[j]
                Insert A[j] into the sorted sequence A[1 . . j - 1].
        i <- j - 1
        while i > 0 and A[i] > key
                do A[i + 1] <- A[i]
                        i <- i - 1
        A[i + 1] <- key
```

For Worst and Best Case: Refer to Question 1(a) of First Term 2012.

(*b*) Floyd Warshall Algorithm

$$A^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \infty & 3 & 8 & -4 & \infty \\ \infty & \infty & \infty & 1 & 1 \\ \infty & 4 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 6 \\ 2 & \infty & -5 & \infty & \infty \end{bmatrix} \end{array}$$

$$A^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \infty & 3 & 8 & -4 & \infty \\ \infty & \infty & \infty & 1 & 1 \\ \infty & 4 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 6 \\ 2 & 5 & -5 & -2 & \infty \end{bmatrix} \end{array}$$

$$A^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \infty & 3 & 8 & -4 & 4 \\ \infty & \infty & \infty & 1 & 1 \\ \infty & 4 & \infty & 5 & 5 \\ \infty & \infty & \infty & \infty & 6 \\ 2 & 5 & -5 & -2 & \infty \end{bmatrix} \end{array}$$

$$A^3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \infty & 3 & 8 & -4 & 4 \\ \infty & \infty & \infty & 1 & 1 \\ \infty & 4 & \infty & 5 & 5 \\ \infty & \infty & \infty & \infty & 6 \\ 2 & 0 & -5 & -2 & 0 \end{bmatrix} \end{array}$$

$$A^4 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \infty & 3 & 8 & -4 & 2 \\ \infty & \infty & \infty & 1 & 1 \\ \infty & 4 & \infty & 5 & 5 \\ \infty & \infty & \infty & \infty & 6 \\ 2 & 0 & -5 & -2 & 0 \end{bmatrix} \end{array}$$

$$A^5 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 4 & 2 & -3 & 0 & 2 \\ 3 & 1 & -4 & -1 & 1 \\ 7 & 4 & 0 & +3 & 5 \\ 8 & 6 & 1 & 4 & 6 \\ 2 & 0 & -5 & -2 & 0 \end{bmatrix} \end{array}$$

Question 4.

(a) Find the optimal parenthesization of a matrix chain product whose sequence of dimensions are <40,30,20,10>.

(b) Determine LCS of <1,0,0,1,0,1,0,1> and <0,1,0,1,1,0,1,1,0>.

Solution:

(a) <40, 30, 20, 10>

$$\left.\begin{array}{l} A_1 : 40 * 30 \\ A_2 : 30 * 20 \\ A_3 : 20 * 10 \end{array}\right\} \; i = 1, j = 3$$

m

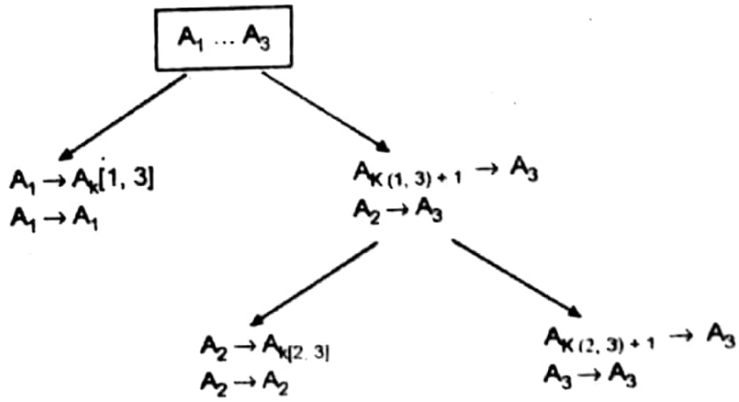
i \ j	3	2	1
1	18000	24000	0
2	6000	0	×
3	0	×	×

k

i \ j	3	2
1	1	1
2	2	0

$i = 1, j = 2$

$$m[1,2] = m[1,1] + m[2,2] + p^0 p^1 p^2$$
$$= 0 + 0 + 40.30.20 = 24000$$
$$m[1,3] = m[1,2] + m[3,3] + p^0 p^2 p^3$$
$$= 24000 + 0 + 40.20.10$$
$$= 32000$$
$$m[1,3] = m[1,1] + m[2,3] + p^0 p^1 p^3$$
$$= 0 + 6000 + 40.30.10$$
$$= 18000$$



Hence optinal parentherization is

$(A_1 \cdot (A_2 \cdot A_3))$.

(b) Refer to anwer 4(a) of first term examination 2011.

□□□

SECOND TERM EXAMINATION

FIFTH SEMESTER [B.Tech.], November, 2016

Algorithm Design and Analysis (ETCS-301)

Time : 1.30 hours *Maximum Marks : 30*

Note: 1. Attempt three questions in total

2. Q. No.1 is compulsory. Attempt any two more question from the remaining.

Question 1.

(a) Differentiate between deterministic and non deterministic algorithms.

(b) Define P and NP class of problems.

(c) Define Greedy Choice Property.

(d) Justify that Naive string matching algorithm uses brute force approach.

(e) Define Matroid.

Solution:

(*a*) Algorithm is deterministic if for a given input the output generated is same for a function. A mathematical function is deterministic. Hence the state is known at every step of the algorithm.

Algorithm is non deterministic if there are more than one path the algorithm can take. Due to this, one cannot determine the next state of the machine running the algorithm. Example would be a random function.

Non deterministic machines that can't solve problems in polynomial time are NP. Hence finding a solution to an NP problem is hard but verifying it can be done in polynomial time. Hope this helps.

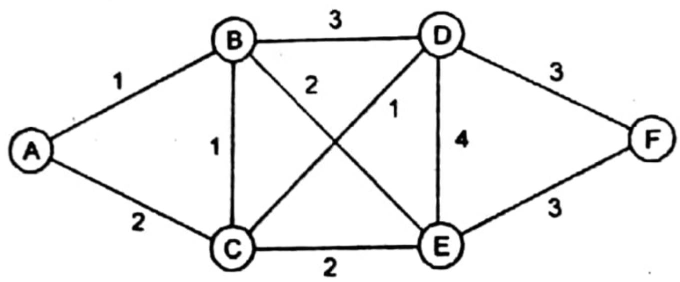
(*b*) Refer to Answer 3(b) of Second-Term Examination 2015.

(*c*) Refer to Answer 1(b) of End Term Examination 2014.

(*d*) In naïve string matching, the pattern is compared with the text at each shift rigorously. Hence, if there are s shifts and the size of pattern is p, then the total number of comparisons are s*p, which is the comparisons of brute force approach. Hence, Naïve SM uses Brute Force Approach.

(*e*) Refer to Answer 1(a) of Second-Term Examination 2010.

Question 2.



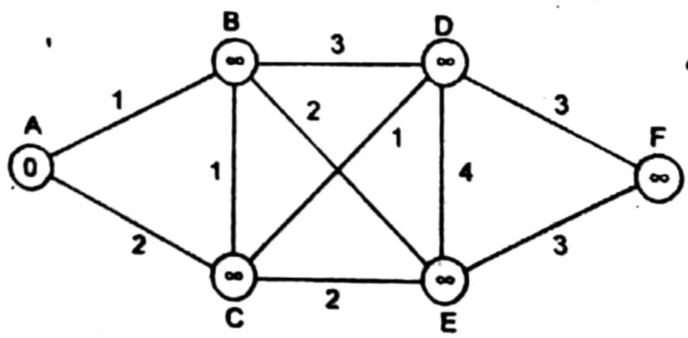
(a) Apply Dijkstra's algorithm on the above graph and find shortest path from node A to rest of the nodes. (Explain step by step).

(b) Find Minimum spanning Tree for the same graph using Prim's Algorithm.

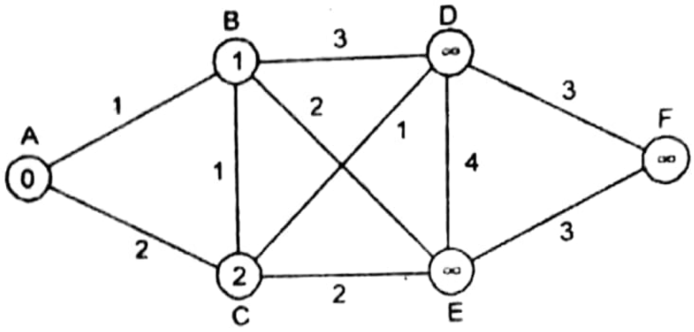
Solution:

(*a*) Dijkstra's algorithm

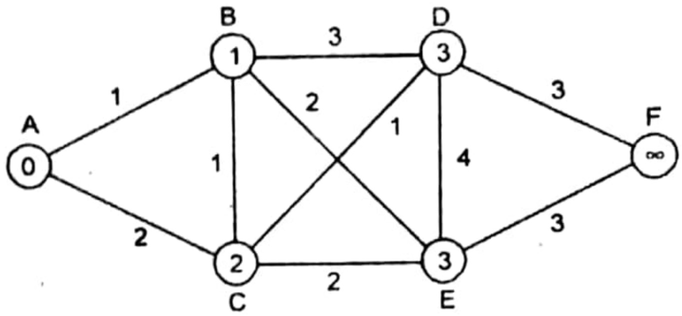
Step I :



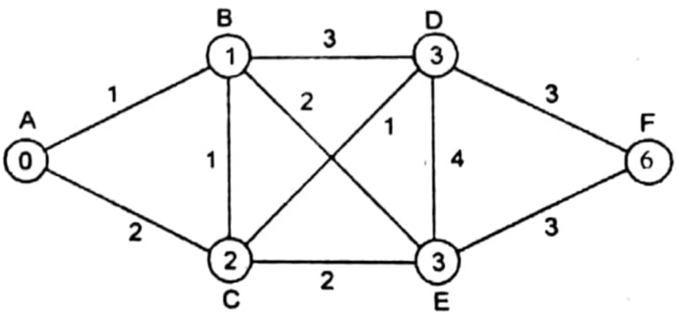
Step II : Update the distance of neighbours of A using relax function (i.e. vertex B and C)



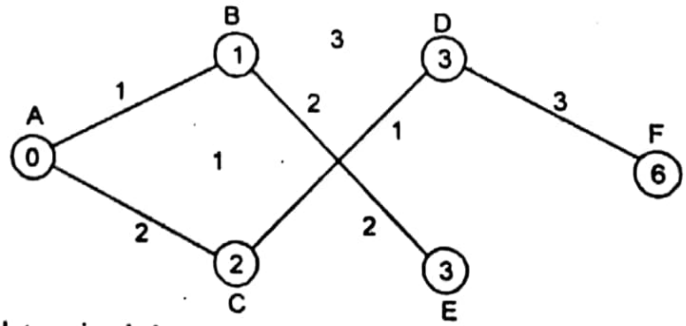
Step III : Now update the distance of neighbours of B and C (i.e. vertex D and E)



Step IV : Now finally update the distance of vertex



Hence the distance graph is :



(b) The graph determined above is also a minimum spanning tree i.e. containing 6 vertex and 5 edges and the graph is connected with total weight = 9.

DiJkstra and prim's algorihtm follow's same approach hence generates same result.

Question 3.

(a) What is activity selection problem, find out the set which contain maximum activity which are compatible to each other:

Ai	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Si	*1*	*2*	*4*	*1*	*5*	*8*	*9*	*11*	*13*
Fi	*3*	*5*	*7*	*8*	*9*	*10*	*11*	*14*	*16*

(b) ***Symbols A,B,C,D,E,F are being produced by information source with protabilities (in percentage) 30,40,6,10,15,4 respectively. Find the binary Huffman code for above symbols.***

Solution:

(a) Activities scheduled is as follows :

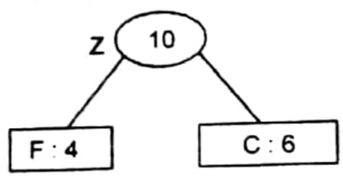
$R <A_1, A_3, A_6, A_8>$

Activity selection problem : we are given '*n*' number of activities, with each activity is associated with start time si and finish time fi we have to schedule all the acitivities such that no two activities are interfare with each other. We are given a resource *R*, which is mutual exclusive i.e if one activity is allocated to *R*, then all other activities will wait. We have to maximize the number of allocated activities.

(b) Max-heap Queue Q_{MH}

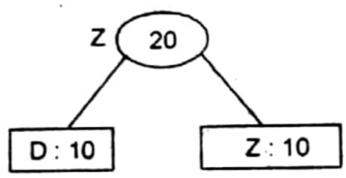
Q_{MH} : <F:4, C:6, D:10, E:15, A:30, B:40>

Step I



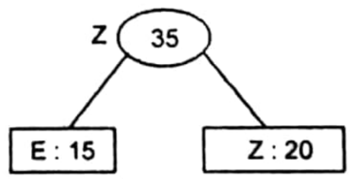
Q_{MH} : <D:10, Z:10, E:15, A:30, B:40>

Step II



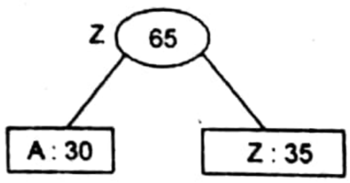
Q_{MH} : <E:15, Z:20, A:30, B:40>

Step III



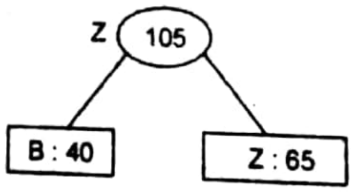
Q_{MH} : <A:30, Z:35, B:40>

Step IV

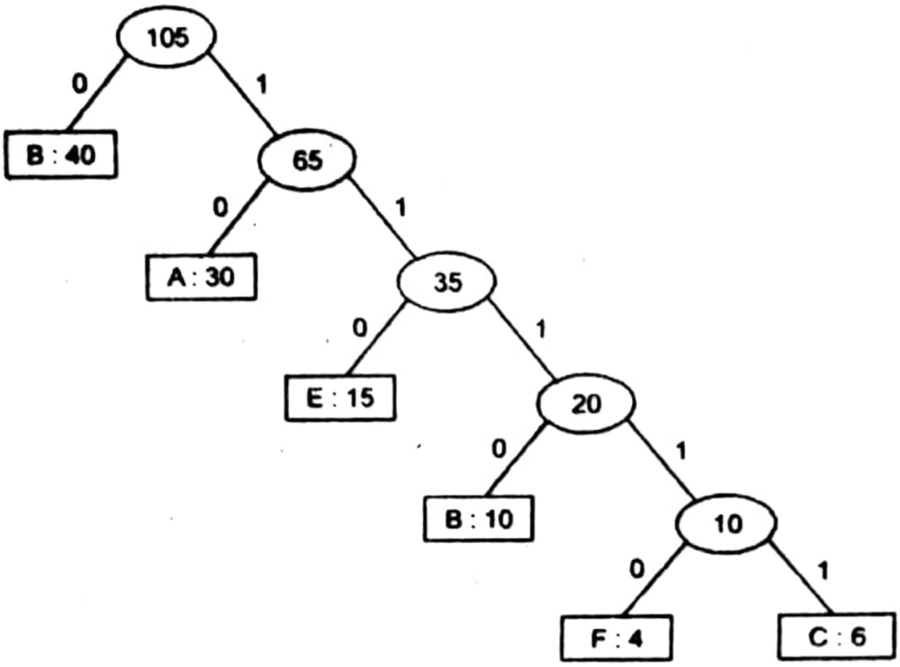


Q_{MH} : <B:40, Z:65>

Step V



Now, the required binary huffman tree



B : 0 D : 1110
A : 10 F : 11110
E : 110 C : 11111

Question 4.

(a)* *Describe Hamiltonian cycle in NP.

(b)* *(i) Write Rabin Karp algorithm

(ii) Solve the given case using Rabin Karp string matching algorithm:

T=3141592653589793 P=26 q=11

Solution:

(*a*) Refer to Answer 9(b) of End-Term Examination 2010.

(*b*) Refer to Answer 3(a) of Second-Term Examination 2015.

□□□

END TERM EXAMINATION

FIFTH SEMESTER [B.Tech.], December, 2016

Algorithm Design and Analysis (ETCS-301)

Time : 3 hours *Maximum Marks : 75*

Note: Attempt any five question including Q.No. 1 which is compulsory. Select one question from each unit.

Question 1.

(a) ***Define little Omega and little Oh notation.***

(b) ***What is recursion tree method?***

(c) ***Write a short note on Memorization.***

(d) ***What is optimal substructure and overlapping substructure?***

(e) ***What are the applications of minimum spanning tree?***

(f) ***Explain fractional knapsack problem.***

(g) ***Differentiate between Dynamic Programming and Greedy Approach.***

(h) ***What is Matroid?***

(i) ***Write Naive String Matching Algorithm.***

(j) ***Explain NP Hard Problem briefly.***

Solution:

(a) **Definition** (Little-o, $o()$): Let $f(n)$ and $g(n)$ be functions that map positive integers to positive real numbers. We say that $f(n)$ is $o(g(n))$ (or $f(n) \in o(g(n))$) if for any real constant $c > 0$, there exists an integer constant $n_0 \geq 1$ such that $f(n) < c * g/(n)$ for every integer $n \geq n_0$.

Definition (Little-omega, $\omega()$): Let $f(n)$ and $g(n)$ be functions that map positive integers to positive real numbers. We say that $f(n)$ is $o(g(n))$ (or $f(n) \in \omega(g(n))$) if for any real constant $c > 0$, there exists an integer constant $n_0 \geq 1$ such that $f(n) < c * g/(n)$ for every integer $n \geq n_0$.

(b) Refer to Answer 2(v) of First-Term Examination 2010.

(c) Refer to Answer 1(b) of First-Term Examination 2010.

(d) Optimal substructure - Refer to Answer 1(d) of End-Term Examination 2011.

Overlapping substructure - Refer to Answer 1(d) of First-Term Examination 2016

(e)
- Consider an application where n stations are to be linked using a communication network.
- The laying of communication links between any two stations involves a cost.
- The problem is to obtain a network of communication links which while preserving the connectivity between stations does it with minimum cost.
- The ideal solution to the problem would be to extract a sub graph termed minimum cost spanning tree.
- It preserves the connectedness of the graph yields minimum cost.

(f) In Fractional Knapsack, we can break items for maximizing the total value of knapsack. This problem in which we can break item also called fractional knapsack problem. An efficient solution is to use Greedy approach. The basic idea of greedy approach is to calculate the ratio value/weight for each item and sort the item on basis of this ratio. Then take the item with highest ratio and add them until we can't add the next item as whole and at the end add the next item as much as we can. Which will always be optimal solution of this problem.

(g) Refer to Q. 1 (b) second term 2010.

(h) Refer to Answer 1(a) of Second-Term Examination 2010.

(i) Refer to Q. 4 (b) second term 2012.

(j) Refer to Q. 8 (b)(i) End term 2010.

Question 2.

(a) Explain Merge Sort and compute the analysis of merge sort.

(b) Perform the Quick Sort to sort the following numbers.

20, 40, 50, 15, 10, 05, 80, 90

Solution:

(a) Merge Sort - Refer to Answer 2(a) of End-Term Examination 2011.

Analysis - Refer to Answer 1(e) of End-Term Examination 2011

(b) Refer to Answer 2(a) of End-Term Examination 2011.

Question 3.

(a) Explain Strassen's algorithm for matrix multiplication.

(b) Apply Strassen's matrix multiplication algorithm to multiply the following matrices.

$$\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} 8 & 4 \\ 6 & 2 \end{bmatrix}$$

Solution:

(a) Refer to Answer 2(b) of End-Term Examination 2011.

(b)

$$\underset{A}{\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}} \underset{B}{\begin{bmatrix} 8 & 4 \\ 6 & 2 \end{bmatrix}} = \underset{Z}{\begin{bmatrix} r & s \\ t & u \end{bmatrix}}$$

where $a_{11}, a_{12}, a_{21}, a_{22}$ are the elements of A and $b_{11}, b_{12}, b_{21}, b_{22}$ the elements of B.

$$P_1 = a_{11}(b_{12} - b_{22}) = 2$$
$$P_2 = (a_{11} + a_{12})b_{22} = 8$$
$$P_3 = (a_{21} + a_{22})b_{11} = 96$$
$$P_4 = a_{22}(b_{21} - b_{11}) = -14$$
$$P_5 = (a_{11} + a_{22})(b_{11} + b_{22}) = 80$$
$$P_6 = (a_{12} - a_{22})(b_{21} + b_{22}) = -32$$
$$P_7 = (a_{11} - a_{21})(b_{11} + b_{12}) = -48$$
$$r = P_5 + P_4 - P_2 + P_6 = 26$$
$$s = P_1 + P_2 = 10$$
$$t = P_3 + P_4 = 82$$
$$u = P_1 + P_5 - P_3 - P_7 = 34$$

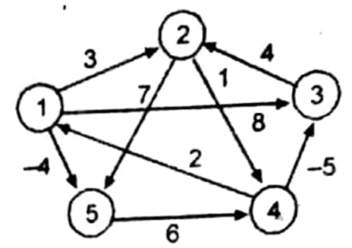
Hence result is

$$Z = \begin{bmatrix} 26 & 10 \\ 82 & 34 \end{bmatrix}$$

Question 4.

(*a*) ***Explain Floyd-Warshall algorithm.***

(*b*) ***Apply Floyd-Warshall algorithm for constructing shortest path.***



Solution:

(*a*) Refer to Answer 7(a)(i) [OR PART] of End-Term Examination 2011

(*b*) Refer to Answer 3(b) First term examination 2017 [only the values of graph has been changed]

Question 5.

(*a*) ***What does Dynamic programming have common with Divide and Conquer and what are differences?***

(*b*) ***Determine the LCS of <A,B,C,B,D,A,B> and <B,D,C,A,B,A>.***

Solution:

(*a*) Refer to Answer 4(a) of End-Term Examination 2011

(*b*) Refer to Answer 2(b) of First-Term Examination 2014

Question 6.

(*a*) ***Explain the difference between Kruskal's and Prim's Algorithm with the help of suitable example.***

(*b*) ***What is an Optimal Huffman Code for the following set of frequencies?***

a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21

Solution:

(*a*) Refer to Q.6 End Term 2012.

(*b*) A Since there are 8 letters in the alphabet, the initial queue size is n = 8, and 7 merge steps are required to build the tree. The final tree represents the optimal prefix code. The codeword for a letter is the sequence of the edge labels on the path from the root to the letter. Thus, the optimal Huffman code is as follows:

h: 0

g: 1 0

f: 1 1 0

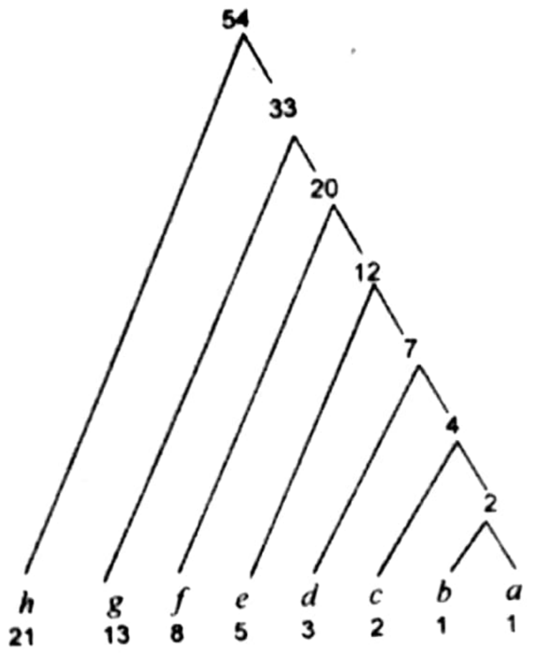
e: 1 1 1 0

d: 1 1 1 1 0

c: 1 1 1 1 1 0

b: 1 1 1 1 1 1 0

a: 1 1 1 1 1 1 1



Question 7.

Illustrate Dijkstra's and Bellman Ford Algorithm for finding the shortest path.

Solution:

Dijkstra Algorithm: Refer to Question 6(a) of End Term 2011.

Bellman Ford Algorithm: Refer Question 7(a)(ii) of End Term 2011 and Question 3(b) of Second Term 2012

Question 8.

(a) Give Knuth Morris Pratt algorithm for pattern matching?

(b) Explain NP-completeness reduction with an example.

Solution:

(a) Refer to Question 9(a) of End Term 2010.

(b) Refer to Question 9 of End Term 2012

Question 9.

(a) Explain Rabin- Karp string matching algorithm.

(b) What is Finite Automata and its significance to match a string with algorithm and complexity.

Solution:

(a) Refer to Question 3(a) of Second Term 2015.

(b) Refer to Question 8 of End Term 2014.

□□□