Indian Institute of Technology, Kharagpur

Date..................... FN/AN | Time: 3 Hrs | Full Marks: 50 | No. of Students: 170

End (Autumn) Sem 2015 | Deptt: MA/EC/CS/IM/HS/BT/EX/CH/ALL

Sub. No. MA 21007 | Subject Name: Design and Analysis of Algorithms

Instruction: Answer **all** questions.

Question 1 [2+2+2+2 = 8 marks]

a) Is the sequence $\langle 20, 15, 18, 7, 9, 5, 12, 3, 6, 2\rangle$ is a max-heap? *Explain.*

b) Where in a max-heap can the smallest element reside, assuming all elements are distinct? Include both the location in the array and the location in the implicit tree structure.

c) Suppose that instead of using `Build-Heap` to build a max-heap in place, the `Insert` operation is used n times. Starting with an empty heap, for each element, use `Insert` to insert it into the heap. After each insertion, the heap still has the max-heap property, so after n `Insert` operations, it is a max-heap on the n elements.

(i) Argue that this heap construction runs in $O(n \log n)$ time.

(ii) Argue that in the worst case, this heap construction runs in $\Omega(n \log n)$ time.

d) If bucket sort is implemented by using heapsort to sort the individual buckets, instead of by using insertion sort as in the normal algorithm, then what are the worst-case and Average case running time of bucket sort? Justify your answer.

Question 2 [$3 + 3 = 6$ marks]

a) Write an algorithm for inserting items in a Red-Black tree. What is the computing time of your algorithm?

b) Start with an empty Red-Black tree and insert the following keys in the given order using your algorithm: $80, 100, 140, 60, 84, 30, 40, 50, 54, 52, 120, 110$

Question 3 [$3 + 3 = 6$ marks]

Consider the graph G on six vertices $\{A, B, C, D, E, F\}$ given by the following adjacency list:

A : $B(4), F(2)$(i.e.A is connected to B with weight 4 and F with weight 2)

B : $A(1), C(3), D(4)$

C : $A(6), B(3), D(7)$

D : $A(6), E(2)$

E : $D(5)$

F : $D(2), E(3)$

a) Describe the order of the vertices encountered on a breadth-first search (BFS) of G starting from vertex A. Break all ties by picking the vertices in alphabetical order (i.e., A before F).

b) Describe the order of the vertices encountered on a depth-first search (DFS) of G starting from vertex A. Break all ties by picking the vertices in alphabetical order (i.e., A before F).

-P.T.O.-

Question 4 [3 + 4 = 7 marks]

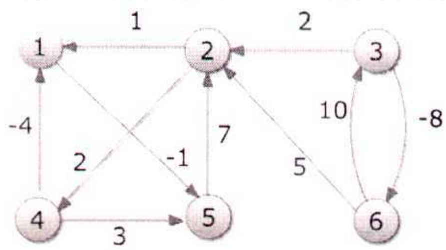
a) Write Bellman-Ford Algorithm for solving shortest path problem.

b) Find a feasible solution or determine that no solution exists for the following system of difference constraints using the Bellman-Ford shortest path algorithm:

$$\begin{aligned}
x_1 - x_4 &\leq -1 \\
x_1 - x_5 &\leq -4 \\
x_2 - x_1 &\leq -4 \\
x_2 - x_3 &= -9 \\
x_3 - x_1 &\leq 5 \\
x_3 - x_5 &\leq 2 \\
x_4 - x_3 &\leq -3 \\
x_5 - x_1 &\leq 5 \\
x_5 - x_4 &\leq 1
\end{aligned}$$

Question 5 [4+3 = 7 marks]

(a) Run the Floyd-Warshall algorithm on the following weighted, directed graph. Show the matrix $D^{(k)}$ that results for each iteration of the outer loop.



(b) How can the output of the Floyd-Warshall algorithm be used to detect the presence of a negative-weight cycle?

Question 6 [16 marks]

TRUE OR FALSE? If the statement is correct, briefly state why. If the statement is wrong, explain why.

a) Given an array $A[1..n]$ of integers, the running time of Heap Sort is polynomial in the input size n.

b) For a dynamic programming algorithm, computing all values in a bottom-up fashion is asymptotically faster than using recursion and memoization.

c) The running time of a dynamic programming algorithm is always $\Theta(P)$ where P is the number of subproblems.

d) In a min-heap, the next largest element of any element can be found in O(log n) time.

e) In a BST, we can find the next smallest element to a given element in O(1) time.

f) Given an unsorted array $A[1..n]$ of n integers, building a max-heap out of the elements of A can be performed asymptotically faster than building a red-black tree out of the elements of A.

g) If a dynamic-programming problem satisfies the optimal-substructure property, then a locally optimal solution is globally optimal.

h) Let $G = (V, E)$ be a directed graph with negative-weight edges, but no negative-weight cycles. Then, one can compute all shortest paths from a source $s \in V$ to all $v \in V$ faster than Bellman-Ford using the technique of reweighting.

-The End-