Institute of Business Administration Karachi
Leadership and Ideas for Tomorrow

Second Term Examination – Spring'18
Max Marks: 20 Duration: 1½ hours

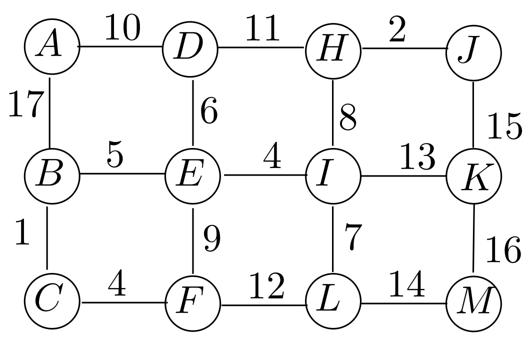
IBA FCS Faculty of Computer Science

Answer all questions. Use seperate answer sheet. Be to the point. Show your work. Please give clear and rigorous answers.

Name: ____________ **ERP:** ________

Question 1: Minimum Spanning Trees .... *7 marks*

(a) [4 marks] Consider the weighted graph below.



i. Run Prim's algorithm starting from vertex A. Write the edges in the order which they are added to the minimum spanning tree. Show the final tree.

ii. Run Kruskal's algorithm. Write the edges in the order which they are added to the minimum spanning tree. Show the final tree.

iii. What is the total weight of minimum spanning tree in this graph?

(b) [1 mark] Can Prim's and Kruskal's algorithm yield different minimum spanning trees? Explain why or why not.

(c) [2 marks] Give a counterexample that shows why the following strategy does not necessarily find the MST: 'Start with any vertex as a single-vertex MST, then add $n-1$ edges to it, always taking next a min-weight edge incident to the vertex most recently added to the MST'

Question 2: True/False .... *5 marks*

For each of the following statements, decide whether it is true or false. Give brief justification of your answer. (1/2 mark for each correct answer and 1/2 mark for each explanation)

(a) $2^{n+1} = O(2^n)$

(b) Consider $f(n) = \log\log n$ and $g(n) = 10^{10^{10^{10^{10^{10}}}}}$. Then $f(n)$ is $O(g(n))$.

(c) Given n integers $a_1, \ldots, a_n$, the third smallest number among $a_1, \ldots, a_n$ can be computed in $O(n)$ time.

(d) If $T_1(n) = O(f(n))$ and $T_2(n) = O(f(n))$, then $T_1(n) = O(T_2(n))$.

(e) The number of comparisons performed by binary search algorithm in worst case is $\Omega(n \log n)$.

Question 3: Divide & Conquer .... *5 marks*

(a) [1½ marks] Here is an alternative array searching algorithm which is a variation of binary search. Instead of checking the middle element of the array (that is the element at position $n/2$) check the element at position $n/3$. Then proceed in the same way as in binary search: if you are looking for x and $A[n/3] = y$ then

- If $x = y$ you have found it.
- If $x > y$ throw away all elements in $A[1 \dots n/3]$.
- If $x < y$ throw away all elements in $A[n/3 \dots n]$.

Write down the recurrence for this algorithm. You do not need to solve your recurrence.

(b) [1½ marks] Solve the recurrence: $T(n) = T(n-1) + 2n$.

(c) [2 marks] Use the divide-and-conquer multiplication algorithm to multiply the two polynomials $P(x) = 7x^3 + 5x^2 + 3x + 1$ and $Q(x) = 2x^3 + 4x^2 + 6x + 8$. (You need to show only one recursion step. Directly multiply the polynomials in the recursive call.)

Question 4: Asymptotic Complexity .... *3 marks*

Algorithms A, B, C, D, E, F, G each solves problem P. For an input of size n,

Algorithm A executes $10n^2$ operations,

Algorithm B executes $\frac{1}{100}2^n$ operations,

Algorithm C executes $100 \log\log(n)$ operations,

Algorithm D executes $20n$ operations,

Algorithm E executes $1000 \log(n)$ operations,

Algorithm F executes $\frac{1}{1000}n!$ operations, and

Algorithm G executes $\frac{1}{10}n\log(n)$ operations.

Rank the algorithms from the fastest to the slowest (i.e. with respect to their asymptotic runtimes)