National University of Computer & Emerging Sciences, Karachi
Fall-2018 Department of Computer Science
Final Exam
27th December 2017, 9:00 am– 12:00 noon

Course Code: CS302	**Course Name:** Design and Analysis of Algorithm
Instructor Name / Names: Dr. Muhammad Atif Tahir, Subhash Sagar and Zeshan Khan	
Student Roll No:	**Section:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **9 questions.**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.

Time: 180 minutes. **Max Marks**: 50 points

Question 1: **(0.5*10=5 Points)**

Answer the following. You should explain in only 3-4 otherwise answer will not be checked.

(a) What is meant by Design and Analysis of Algorithms?
(b) List two topics in Computer Science that are more important than studying computer program performance.
(c) Write down the formal definition of Big-Oh Notation i.e. in terms of f(n) and g(n)
(d) List one condition / scenario where Master Theorem cannot be applied.
(e) What are the average running time complexities of Insertion and counting sort?
(f) List two algorithms that belong to Divide and conquer domain.
(g) List two algorithms that belong to Greedy Algorithm domain.
(h) List two algorithms that belong to Dynamic Programming domain.
(i) What is the complexity of Polynomial time algorithms?
(j) What are the limitations of Greedy Algorithms?

Solution 1: **(0.5*10=5 points)**

(a) The analysis of algorithm is the theoretical study of computer program performance and resource usage. Algorithm design include creating an efficient algorithm to solve a problem in an efficient way using minimum time and space.
(b) Correctness, Security, Stability etc
(c)

$$O(g(n)) = \{f(n) : \text{there exists positive constants } c \text{ and } n_0 \text{ such that } f(n) \leq cg(n) \text{ for all } n \geq n_0\}$$

(d)

- $T(n)$ is not monotone, ex: $T(n) = \sin n$
- $f(n)$ is not a polynomial, ex: $T(n) = 2T(\frac{n}{2}) + 2^n$
- b cannot be expressed as a constant, ex: $T(n) = T(\sqrt{n})$

(e) $O(n^2)$ for insertion sort, $O(n+k)$ for counting sort
(f) Merge Sort, Quick Sort
(g) Dijkstra, Kruskal
(h) Matrix Chain, 0/1 Knapscak
(i) Polynomial-time: running time is $O(n^k)$, where k is a constant.
(j) Stuck in local minima.

Question 2: **(2+2=4 points)**

Recurrence Equation of Quick Sort is $T(N) = T(i) + T(N - i - 1) + cN$. For worst case, pivot point is the smallest element while for best case, pivot is the middle element. Show by using either substitution or recurrence / iterative method that the worst-case complexity is $O(n^2)$ and best-case complexity is $O(nlogn)$.

Solution 2: **(2+2=4 points)**

A) Worst case analysis

The pivot is the smallest element
T(N) = T(N-1) + cN, N > 1
Telescoping:
T(N-1) = T(N-2) + c(N-1)
T(N-2) = T(N-3) + c(N-2)
T(N-3) = T(N-4) + c(N-3)
T(2) = T(1) + c.2
Add all equations:
T(N) + T(N-1) + T(N-2) + … + T(2) = T(N-1) + T(N-2) + … + T(2) + T(1) + c(N) + c(N-1) + c(N-2) + … + c.2
T(N) = T(1) + c(2 + 3 + … + N)
T(N) = 1 + c(N(N+1)/2 -1)
Therefore T(N) = O(N^2)

B) Best case analysis

The pivot is in the middle
T(N) = 2T(N/2) + cN
Divide by N:
T(N) / N = T(N/2) / (N/2) + c
Telescoping:
T(N/2) / (N/2) = T(N/4) / (N/4) + c
T(N/4) / (N/4) = T(N/8) / (N/8) + c
……
T(2) / 2 = T(1) / (1) + c
Add all equations:
T(N) / N + T(N/2) / (N/2) + T(N/4) / (N/4) + …. + T(2) / 2 = (N/2) / (N/2) + T(N/4) / (N/4) + … + T(1) / (1) + c.logN
After crossing the equal terms:
T(N)/N = T(1) + cLogN = 1 + cLogN
T(N) = N + NcLogN
Therefore T(N) = O(NlogN)

Question 3: **(3+2=5 points)**

a) What is meant by P and NP Problems? Explain P = NP [3 Points]

b) A problem that is solvable in time complexity of $T(n) = 3 * n^n$ and space complexity of $S(n) = n^2$ and it can be validated in $T(n) = 2^n$ time. Is it a NP-Complete or NP-Hard?

Solution 3: **(3+2=5 points)**

A)

- P problems
  - (The original definition) Problems that can be solved by **deterministic Turing machine** in polynomial-time.
  - (A equivalent definition) Problems that are solvable in polynomial time.
- NP problems
  - (The original definition) Problems that can be solved by **non-deterministic Turing machine** in polynomial-time.
  - (A equivalent definition) Problems that are **verifiable** in polynomial time.
    - Given a solution, there is a polynomial-time algorithm to tell if this solution is correct.

P = NP means whether an NP problem can belong to class P problem. In other words, whether every problem whose solution can be verified by a computer in polynomial time can also be solved by a computer in polynomial tim

B) NP-Hard as validated in $T(n) = 2^n$ time.

Question 4: **(3 points)**

For TSP graph, Proof by Contradiction that Weight of $MST \leq OPT$ where OPT denotes the cost of the minimum weight tour in TSP graph.

Solution 4: **(3 points)**

Proof By contradiction: Assume an instance of a Metric TSP with $OPT < MST$. Removing an edge from OPT creats an acyclic graph hitting every node once, therefore it is a spanning tree with weight $T < OPT$, so $T < OPT < MST$. Therefore MST was *not* minimal, a contradiction. ■

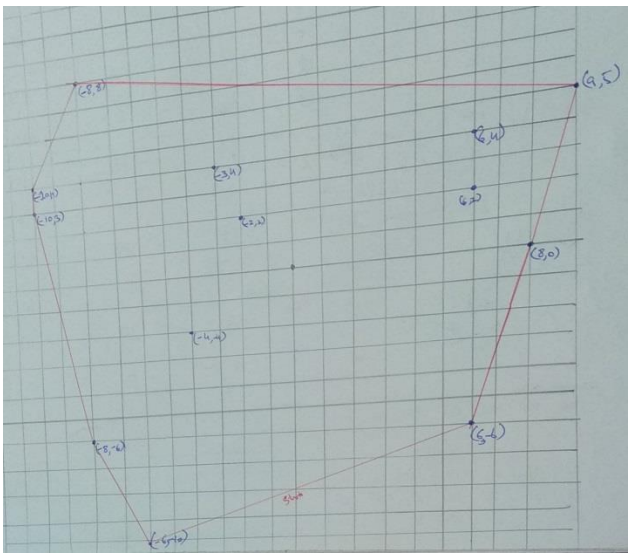
Question 5: **(6 points)**

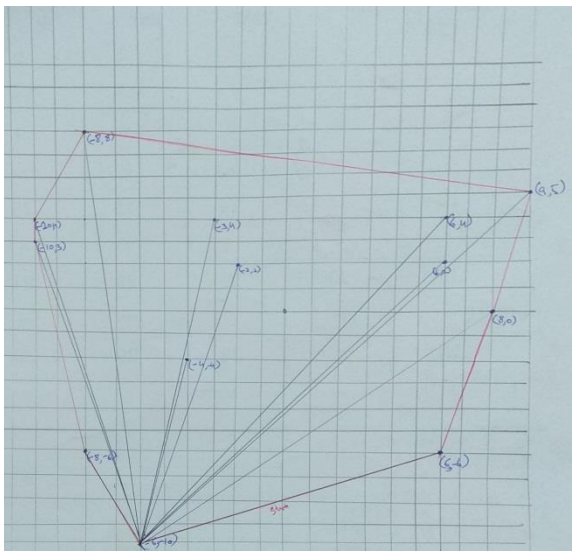
Consider following points in 2D
(6,2), (9,5), (-2,2), (-3,4), (-8,8), (-10,4), (-10,3), (-8,-6), (-4,-4), (6,4), (6,-6), (-6,-10), (8,0)
Find the smallest convex set containing all the points using Package Wrap (Jarvis March) and Graham Scan (Show all iterations). Pseudo code of both algorithms is given in Appendix 1.

Solution 5: **(6 points)**

A) Jarvis March: The points at the hull of convex in ordered form are: $(-6,-10) \rightarrow (6,-6) \rightarrow (8,0) \rightarrow (9,5) \rightarrow (-8,8) \rightarrow (-10,4) \rightarrow (-10,3) \rightarrow (-8,-6)$.

B) Graham Scan: The points are as same as the Jarvis March but the deleted points are: $(6,2) \rightarrow (6,4) \rightarrow (-2,2) \rightarrow (-4,-4) \rightarrow (-3,4)$. The sorted points are: $(-6,-10) \rightarrow (6,-6) \rightarrow (8,0) \rightarrow (9,5) \rightarrow (6,2) \rightarrow (6,4) \rightarrow (-2,2) \rightarrow (-4,-4) \rightarrow (-3,4) \rightarrow (-8,8) \rightarrow (-10,4) \rightarrow (-10,3) \rightarrow (-8,-6)$.





Question 5.B: **(5 points)**

Given a list of ***n-1*** integers and these integers are in the range of 1 to ***n***. There are no duplicates in list. One of the integers is missing in the list. Design O(n) running time algorithm to find the missing integer.

Solution 5.B: **(5 points)**

```
#include<stdio.h>
/* getMissingNo takes array and size of array as
arguments*/
int getMissingNo (int a[], int n)
{
    int i, total;
    total  = (n+1)*(n+2)/2;
    for ( i = 0; i< n; i++)
       total -= a[i];
    return total;
}
/*program to test above function */
int main()
{
    int a[] = {1,2,4,5,6};
    int miss = getMissingNo(a,5);
    printf("%d", miss);
    getchar();
}
```

Question 6: **(2+3=5 points)**

We are interested to send some of the students to visit different universities in the city and let them know about Procom (Fast University Event). There are "*n*" universities and *n* students are selected (1 for each university). The sample map of the universities is available below with the point "A" as the main campus of FAST where the students are waiting to start visit.

A. Design a time optimal algorithm within $O(n^2)$ to compute the shortest path for all students from A to visit all universities.

B. Apply your algorithm on the provided example graph in "**Figure: 1"** and write the shortest path for each of the student from A.

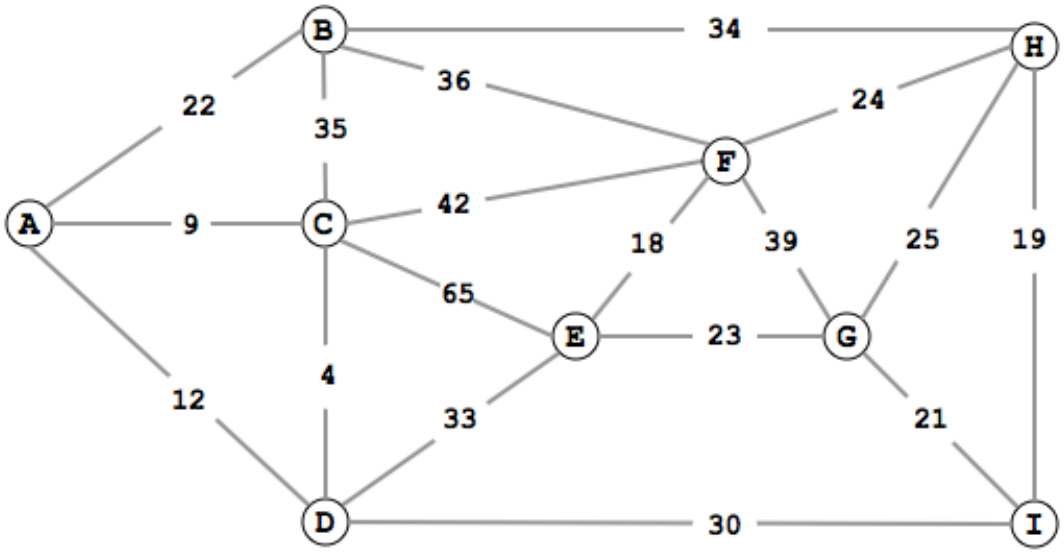


Figure: 1

Solution 6: **(2+3=5 points)**

A) Dijkstra Algorithm or Its pseudo code
B) The final output after execution of the Dijkstra Algorithm is:

	A	B	C	D	E	F	G	H	I
Final	0,Nil	22,A	9,A	12,A	45,D	51,C	63,I	56,B	42,D

Question 7: **(6 points)**

Suppose that we have a given set of five matrices A1, A2, A3, A4 and A5 with the following dimensions:

Matrix	A1	A2	A3	A4	A5
Dimension	5 X 2	2 X 3	3 X 1	1 X 4	4 X 3

Use the Matrix Chain Multiplication algorithm to discover the optimal parenthesization of the matrices with the cost of multiplication.

Solution 7: **(6 points)**

$\big((A1(A2A3))(A4A5)\big)$

Cost Matrix

	A1	A2	A3	A4	A5
A1	0	30	16	36	43
A2		0	6	14	24
A3			0	12	21
A4				0	12
A5					0

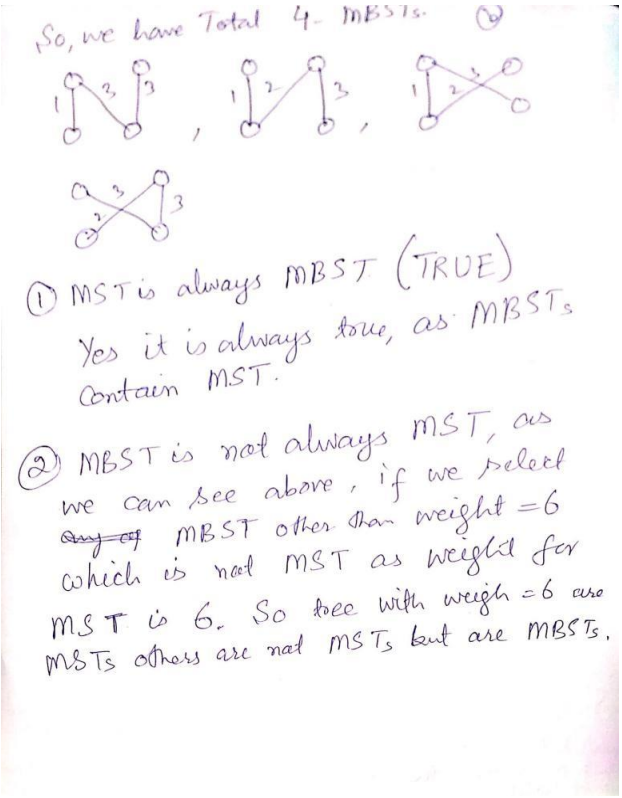
Question 8: **(6 points)**

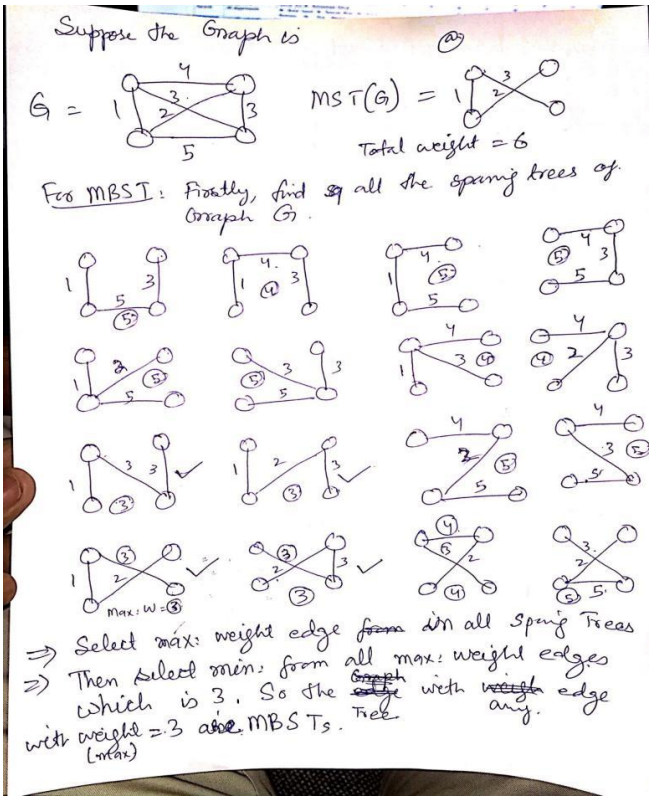
Minimum Bottleneck Spanning Tree (MBST) in an undirected graph is a spanning tree in which the most weighted edge is as cheap as possible among all the possible combinations of spanning trees. A bottleneck edge is the highest weighted edge in a spanning tree. A spanning tree is minimum bottleneck spanning trees if the graph contains a spanning tree with a minimum bottleneck edge weight. More clearly, for a graph; list all the possible spanning trees. Among these pick the spanning tree whose maximum edge weight is minimum.

Prove or Disprove with example using graph? (Graph must be of at least 4 vertices).

1) Whether a MST is always a MBST?
2) A MBST is always a MST?

Question 8: **(6 points)**





Question 9: **(5 points)**

For a graph given in **Figure: 2**, Solve the vertex cover problem using the following two methods.

1) 2 approximation greedy algorithm for vertex cover problem
2) Using the following algorithm
   "To solve the vertex-cover problem. Repeatedly select a vertex of highest degree, and remove all of its incident edges."

For a given graph, do both the algorithms give the same result or not.

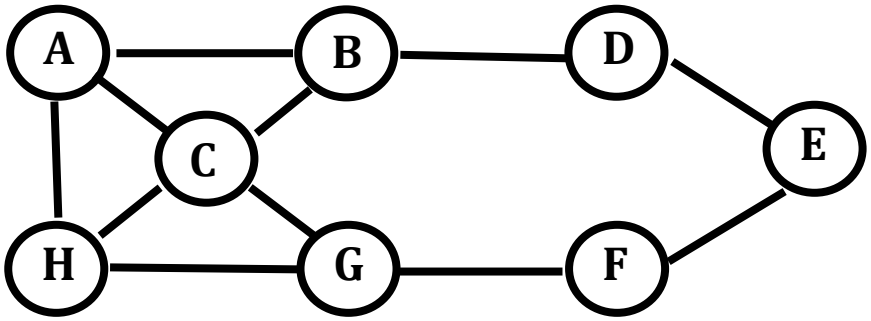


Figure: 2

Solution 9: **(5 points)**

Q No. 9 (Solution)

Part: (1) Possible Solution

(i) Select edge from node: A to node: B
C = {A, B}

(ii) edge from node: C to node: G
C = {A, B, C, G}

(iii) edge from node: D to node: E
C = {A, B, C, G, D, E}

Part (2)

(i) Select node: C
Array = {C}

(ii) All the remaining nodes ~~other than E~~ can be selected as all the nodes have same degree
⇒ if we select node: A.
Array = {C, A}

(iii) Three possible node can be selected from G, F and D.
⇒ if we select node: G
Array = {C, A, G}

(iv) ~~Only one~~ Two possible selection either D or E
Array = {C, A, G, E}

(v) Select D or B.
Array = {C, A, G, E, D}

BEST OF LUCK

Appendix 1:

Package wrap.

- Start with point with smallest y-coordinate.
- Rotate sweep line around current point in ccw direction.
- First point hit is on the hull.
- Repeat.

Graham scan.

- Choose point p with smallest y-coordinate.
- Sort points by polar angle with p to get simple polygon.
- Consider points in order, and discard those that would create a clockwise turn.