

National University of Computer and Emerging Sciences

Lahore Campus

CLO #1: Implement the algorithms, compare the implementation with the fundamental algorithms knowledge to have practical problems related to the CLO.

Q2) It is observed in NUCES that our LAN is not properly designed and is not using the available resources to its fullest. As you people are familiar with the fact that the NUCES Lahore campus consists of various building blocks so the LAN needs to connect the classrooms, labs, faculty offices in these buildings efficiently.

We have modeled this problem to a graphs, where computers in classrooms, labs, offices are represented by nodes and the networking wires that connect them are represented by edges. The edges have weight that indicates the length of the wire needed to connect two nodes. The management wants to design a LAN network, FLAN that covers all the nodes with minimum wire length and do so efficiently.

length and do so efficiently.

- a. Which graph algorithm (that you have already studied in class you think is most suitable to compute the FLAN and what is time complexity in term of $|V|$ and $|E|$?)
- b. The ideal FLAN may not be possible due to various issues. For example, the optical fiber might need underground wiring which means additional cost even if the length of the required wire is short. After providing the program that produces the best FLAN, you are now asked to add the functionality that will help the director's office to modify the networking by choosing to disallow a certain connection in the existing network. When a connection is disallowed, the FLAN cannot use that connection anymore. Your algorithm should update the best network accordingly. It should take linear time to do so i.e $O(|V|+|E|)$.
- c. A recent technology has been introduced and it has been estimated that a direct connection between any two nodes in the network can in fact be cabled in \$5000, regardless of the distance between them. This means that the cost of each connection becomes a fixed amount of \$1000. Additionally, k of these connections have been sponsored by a software house. Without even computing the ideal network, can you tell what could be the minimum possible cost of the network under the new scenario that FAST needs to pay?

CLO #1: Design algorithms using different algorithms design techniques i.e. Brute Force, Divide and Conquer, Dynamic Programming, Greedy Algorithms and apply them to solve problems in the domain of the program

Q3:

- Given a binary string S of type 1^m0^n , devise an algorithm that finds the number of zeroes in $O(\log k)$ time. Given a binary string S of type 1^m0^n , devise an algorithm that finds the number of zeroes in $O(\log k)$ time. Here $m+n=k$. Write 3-4 lines to explain your idea.

Binary String = {111100000000}

We use divide and conquer approach to divide the string. Here upon dividing we get a zero so we have at least 1 zero. We can have more so we divide the string again. If we get 0 again we repeat and add one with our answer. In our answer we stop when we get 1, then we divide the right substring and repeat the recursive process.

Spring 2024

School of Computing

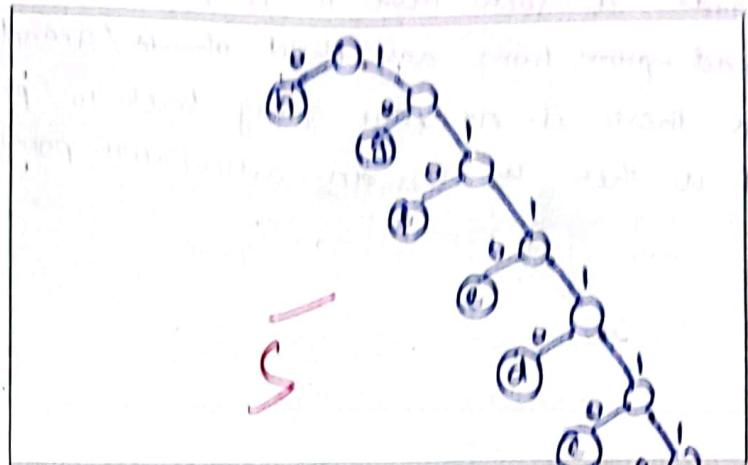
Page 2 of 6

right substring and repeat the recursive process

```
zeroes(&S, l, r){  
    m = l + r  
    if S[m] == '0'  
        return 1 + zeroes(&S, l, m);  
    else  
        return zeroes(&S, m + 1, r);  
}
```

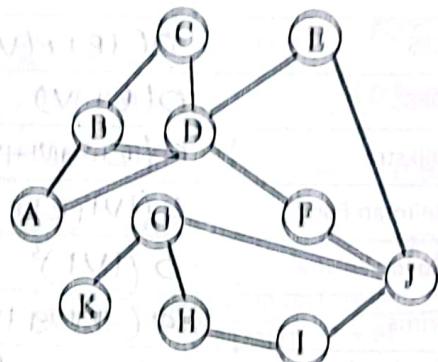
- b. Few days back I generated these codes using Huffman coding. Unfortunately, I lost the paper where I drew that tree. Please help me in re-generating the tree using given codes

Character	Code
a	1111110
b	1111111
c	111110
d	11110
e	1110
f	110
g	10
h	0



- c. Suppose you are a medical practitioner and it is your job to vaccinate people in each village of Pakistan to limit the spread of adenovirus in the country. The adenovirus can be transmitted between two people if they handshake. For each village, you are given a handshaking graph, G , whose vertices are the people in that village and whose edges are pairs of people who regularly handshake. You have limited supply of the vaccine, and you must have to vaccinate the central hand shakers only, where in the graph a central hand shaker is a person D , such that there are no two people C and E , who are hand shake by D such that there is a sequence of handshaking pairs of people that starts with C and leads to E while avoiding D .

- Which vertices are the central hand shakers in this graph?
- Describe an efficient algorithm for identifying all the central hand shakers in G , and provide its running time. Explain your idea in 3-4 lines.



(i) J, G, D are central hand shakers

(ii) we need to find articulation points. We can use modified DFS to compute the low value for every vertex $\text{low}[v] = \min(d[v], \text{lowest_low}[v])$ from tree edges, lowest $d[u]$ in back edges (if there exists a node u such that for a child v $\text{low}[v] \geq d[u]$ then u is articulation point / central hand shaker.)

Running time is $O(|E| + |V|)$

central hand shaker.

or we can say in DFS tree if there exist a node u such that it is not root of the tree and parent from any child of u/grandchild of u then it no edge going back to proper ancestor of u then u is an articulation point/central hand shaker.

Q10.2: Analyze the time and space complexity of different algorithms by using standard asymptotic notations for recursive and non-recursive algorithms.

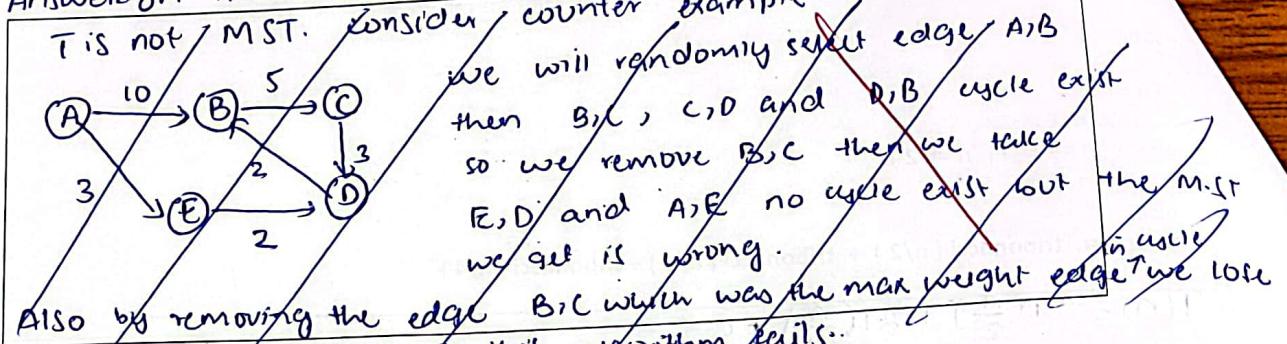
W3

[Marks: 5 + 5]

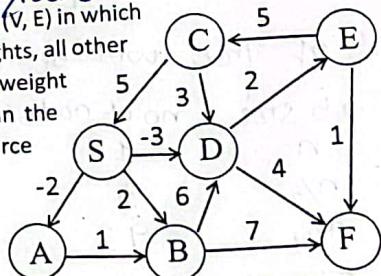
a. Fill the Table. Also mention the choice of data structure for graph algorithms.

Algorithm Name	Worst Case Time Complexity
BFS	$O(E + V)$ ✓ using Queue DS
DFS	$O(E + V)$ ✓ after visit adjacency list for graph
Dijkstra	$O(E \log V + V \log V)$ ✓ Priority Queue
Bellman Ford	$O(V E)$ ✓ adjacency list for Graph
Floyd Warshall	$O(V ^3)$ ✓ adjacency list for Graph
Prims	$O(E \log V)$ ✓ Priority Queue
Kruskal	$O(E \log V)$ ✓ Priority Queue
Activity Selection Problem	$O(n^2)$ ✗
Huffman Coding	$O(V \log V) \cdot n \log n$ ✓ using MinHeap
0-1 knapsack Problem	$O(n^2)$ / $O(W * I)$: W=total weight, I=no. of items
Fractional Knapsack Problem	if sorted already then $O(n)$ otherwise, $n \log n$
Edit Distance	$O(n^2)$ / $O(n_1 \times n_2)$ n ₁ =length of string 1 n ₂ =length of string 2
Rod Cutting	$O(n^2)$ / $O(L^x \text{ pieces})$ L=length of rod pieces, possible pieces we can cut in

Answer on Answer Sheet.



- b. Suppose that you are given a weighted, directed graph $G = (V, E)$ in which edges that leave the source vertex may have negative weights, all other edge weights are nonnegative, and there are no negative-weight cycles. An example of such graph is provided below. Can the Dijkstra's algorithm correctly find shortest paths from source S? Provide the valid justification.



Yes, such algorithm will provide shortest path.

Problem occurred when we took Djikstra and a node could not be relaxed because we already relaxed it and the negative edge did not contribute to shortest path in final answer. Here Djikstra will first account for negative weight edge and relax the nodes connected to source since they get relaxed then there is no chance we will miss out on any relaxation.



National University

of Computer & Emerging Sciences

A
16/24

SEMESTER : SPRING SUMMER FALL Year 2024

Please Tick Appropriate Boxes

Course Design & Analysis of Algorithms Serial No./ _____

05352

Roll No. 221-0504 Section BBS-4A Date _____

Serial No. of continuation sheet(s) attached _____

INSTRUCTIONS FOR CANDIDATES

- Write Question No. In the middle of the line using thick tipped pen.
- Use only blue or black pen to write your answers.
- Answers written using pencil will not be checked.
- Pencil is only allowed to draw diagrams or write program code.

(THIS ANSWER BOOK CONTAINS NOS. 1-18)

Q./Part No.	Marks
Q. - 1	
Q. - 2	
Q. - 3	
Q. - 4	
Q. - 5	
Q. - 6	
Q. - 7	
Q. - 8	
Q. - 9	
Q. - 10	

Q./Part No.	Marks
Q. - 11	
Q. - 12	
Q. - 13	
Q. - 14	
Q. - 15	
Q. - 16	
Q. - 17	
Q. - 18	
Q. - 19	
Q. - 20	

Marks Obtained

Total Marks

Examiner's Signature

Date

Art No.

QUESTION 01

Recursive

$$\text{MaxProfit}(t, b, n) = \max \begin{cases} \text{maxProfit}(t, b, n-1), \\ \text{maxProfit}(t-t_n, b, n-1) + b_n \end{cases} \quad \text{if } T_n \leq t$$

$$= \text{maxProfit}(t, b, n-1) \quad \text{if } T_n > t$$

We make two calls for the MaxProfit

for one call we pass

$$\text{MaxProfit}(t, b, T, N)$$

for other we pass

$$\text{MaxProfit}(t, b, T, N-1)$$

The max of these two results will be our final answer.

Bottom-up Solution

$$\text{MaxProfit}(t, b, T, N) \{$$

~~init / dp[0][t][0:N]~~~~for (i=0 to T)~~

~~dp[i][0] = 0~~

~~for (i=0 to N)~~

~~dp[0][i] = 0~~

~~for (i=1 to T)~~~~for (j=1 to N) {~~~~if (T < t[j]) {~~

~~dp[i][j] = dp[i-1][j];~~

~~} else {~~

~~dp[i][j] = max (dp[i-t[j]][j-1] + b_j, dp[i-1][j]);~~

```


$$\begin{cases}
    \text{if } (i=1) \\
    \quad \text{if } (T < t[i]) \\
    \quad \quad dp[i][j] = 0 \\
    \quad \text{else } dp[i][j] = b[i];
\end{cases}$$


```

not

Max Profit (b , t , T , N) :

```
int dp[0:N][0:T];
```

```
for (i=0 to N)
```

```
    dp[i][0] = 0
```

```
for (j=0 to T)
```

```
    dp[0][j] = 0
```

```
for (i=1 to N)  $\leftarrow$  increment i by 2
```

```
    for (j=1 to T) {
```

#1

```
    \leftarrow else {  $\leftarrow T < t[i]$ 
```

```
        if ( $T < t[i]$ )
```

```
            dp[i][j] = dp[i-2][j];
```

```
        else {
```

```
            dp[i][j] = max( dp[i-2][j],
```

```
            dp[i-2][j - t[i]] + b[i] )
```

}

}

✓

```
for (i=2 to N) (increment i by 2)
```

```
    for (j=2 to N) {
```

```
        if ( $T < t[i]$ )
```

```
            dp[i][j] = dp[i-2][j];
```

```
        else {
```

```
            dp[i][j] = max( dp[i-2][j],
```

```
            dp[i-2][j - t[i]] + b[i] )
```

}

}

return $\max(\text{dp}[N][T], \text{dp}[N-1][T/2])$;

} Time Complexity = ~~$O(T^N)$~~ $O(T^N)$

first nested loops fills half of dp array the
other nested loop fills other half.

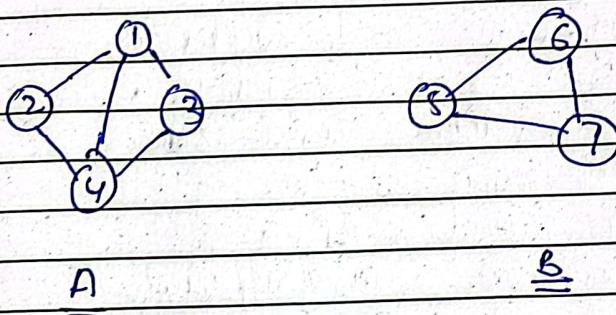
Question 02

(a) Prims algorithm for MST.

Kruskals Algorithm for MST

Time complexity = $O(|E| \log |V|)$

(b) By removing one node we get two separate connected components say A and B.



We can run DFS on A and color all the nodes black.

Now we run DFS on any node of B, in G.

In DFS-visit, if we encounter any node with color black we store that edge in minEdge variable. We will update minEdge variable if we encounter any other edge such that one node is white and other is black & its weight < minEdge.

Finally we get the minimum edge after DFS is complete. This minimum edge \neq the one rem-

and joins two components back together.
This takes $O(|V| + |E|)$ time.
The new MST = oldMST \cup minEdge

10

(c)

If there are $|V|$ nodes
then FAST will need to
pay

$(|V|-1-k) \times 1000$ \$ for
the connection

3

QUESTION - 05

(a)

The tree will be a Minimum Spanning Tree.

This method is similar to Kruskal's Algorithm where we pick edges in increasing order of their weight and don't allow any cycle to occur.

Here instead of picking edges in increasing order we arbitrarily chose edges and if cycle occur we remove the edge in cycle with max weight.

This approach works because we are always removing the edge of cycle with maximum weight.

Proof by counter example

Say that T we get is not M-S-T if T is not M-S-T then there

exists an M-S-T T' such that T' contains an edge with less weight than one in T .

This means T contains an edge with

Part No.

weight greater but we already removed all edge forming edges in T with max weight. we made optimal choice at occurrence of every cycle.
Hence T is a MST.

DFS - VISIT

time_cd[V] = time_cnew_c 0