A

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Final Exam | Semester: Fall 2024
Duration: 1 Hour 40 Minutes | Full Marks: 40

CSE 221: Algorithms

Answer the following questions.
Figures in the right margin indicate marks.

Name:	ID:	Section:

1 You have just joined a new university and are exploring the campus, which consists of several buildings connected by one-way bridges. These bridges form a directed graph, where each node represents a building and each edge represents a bridge between two buildings. You aim to analyze how the buildings and bridges are connected to understand the campus layout better and improve your navigation between classes and facilities.

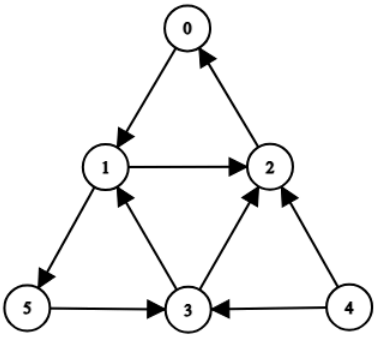


Figure 1

CO2 a) **Mention** which graph representation will let you check whether any two buildings have a bridge between them in O(1) time. **01**

b) From Figure 1

CO1 i. **Simulate** an algorithm to find the Strongly Connected Components (SCC). **04**

CO3 ii. Suppose a bridge gets blocked. **Explain** how you can efficiently calculate the updated SCCs from b(i) without rerunning the algorithm on the entire graph. **02**

CO3 c) A building is called "*Accessible*" if it's possible to start from there, visit some (or all) of the other buildings, and eventually return to that building. **03**

Propose an algorithm with a pseudocode/programmable code/step-by-step logical instructions to determine whether every building on the campus is *Accessible*. The algorithm should run in **at most O(V × (V + E))** time.

2 Imagine, you and your best friend are trapped in a dungeon, trying to escape quickly. The dungeon is made up of chambers connected by two-way tunnels. However, these tunnels may contain dangers that drain your energy fighting it. You have a map showing how much energy you will require while passing through each of the tunnels. The energy requirements for each tunnel are not necessarily the same.

Now answer the following questions.

a. CO1 You are given the map as an Adjacency List where the chambers are numbered from 1 to N. You are in chamber 1 and aiming to reach chamber N using a path that requires minimum energy. **02**

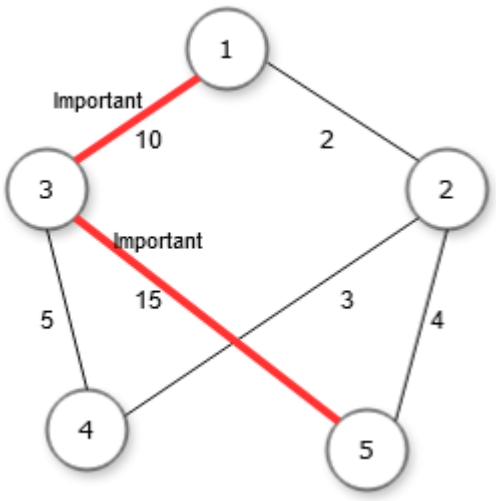
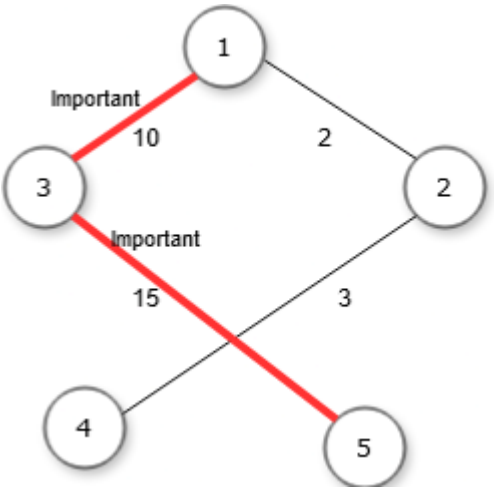
Name the algorithm you prefer to solve this problem. **Explain** your choice.

b. CO2 If some tunnels contain an energy drink that will *restore* your energy by a certain amount, will your proposed algorithm in (a) work? **Explain** Briefly. **01**

c. CO2 You realize that you forgot some of your belongings in some tunnels. Hence, you want to construct the MST such that those tunnels are <u>always</u> included no matter how dangerous they are. **04**

Formally, you have an undirected weighted graph with some edges marked as "*important*" and you need to propose a modification to Kruskal's algorithm such that you can find the MST that always includes those important edges (even if it increases the total cost).

For simplicity, you can assume the important edges/tunnels do not create any cycles. Look at the example below.

Input Graph	Modified Minimum Spanning Tree
	

Present your algorithm with a programmable code/ pseudocode/ step-by-step instruction.

d. **CO3** The map of the island is too big to carry. Thus, your friend came up with an idea to use Kruskal's Algorithm to only store the Minimum Spanning Tree (MST) of the graph. He stated that "*The minimum energy needed to travel between any two chambers in the MST is the same as in the Original Map*". Do you agree with the statement? **Justify** your answer with proper examples and necessary details. **03**

3 Consider the following scenario

You are designing a data compression system for a critical database server. You have analyzed the following character frequencies from the database logs:

Character	Frequency
A	45
B	13
C	12
D	16
E	9
F	9
G	12

Now answer the following questions:

a. **CO1** **Construct** a Huffman tree for the given characters and their frequencies. Additionally, calculate the space saved compared to using fixed 3-bit encoding for each character. Show your work step by step. **04**

b. **CO2** **Write** the Huffman Code for each character. **02**

c. **CO2** **Explain** how greedy property is used in Huffman coding and how it provides an optimal code. **02**

d. **CO2** Two friends are tasked with building the given Huffman tree. While both friends use priority queues in the process, **02**

- friend 1 inserts the newly created nodes at the end of the same frequency nodes.
- friend 2 inserts the newly created node at a random position between the same frequency nodes.

Explain which technique will provide more optimal code and why. If they are bound to provide equally optimal code, **explain** why.

4 Answer the following questions:

a. Can merge sort be written as a dynamic programming algorithm? **Explain** your answer. **02**
CO2

b. We are calculating the longest common subsequence (LCS) between two strings, $S = X_1X_2X_3X_4$ and $T = Y_1Y_2Y_3Y_4$. To do this, we fill an array C where $C_{i,j}$ represents the length of the LCS between the prefix of length i from S and the prefix of length j from T.

Some of the entries in the array C are currently masked using different symbols other than integers as shown below.

	Y_0	Y_1	Y_2	Y_3	Y_4
X_0	0	0	0	0	0
X_1	0	0	0	0	1
X_2	0	0	0	1	1
X_3	0	0	1	1	♡
X_4	0	1	♣	♠	♢

CO1
i. What does the cell $C_{0,0}$ represent? **01**
ii. What can be said about the characters X_1 and Y_4? **01**
iii. What can be said about the characters X_2 and Y_4? **01**
iv. What is the value of ♡? **01**
v. What is the value of ♣? **01**
vi. When we are filling up the i-th row of our dynamic programming table C, what rows do we need to have access to? **01**
CO3 vii. Given the observation above can we optimize our space complexity further? **02**

B

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Final Exam | Semester: Fall 2024
Duration: 1 Hour 40 Minutes | Full Marks: 40

CSE 221: Algorithms

Answer the following questions.
Figures in the right margin indicate marks.

Name:	ID:	Section:

1 You have just joined a new university and are exploring the campus, which consists of several buildings connected by one-way bridges. These bridges form a directed graph, where each node represents a building and each edge represents a bridge between two buildings. You aim to analyze how the buildings and bridges are connected to understand the campus layout better and improve your navigation between classes and facilities.

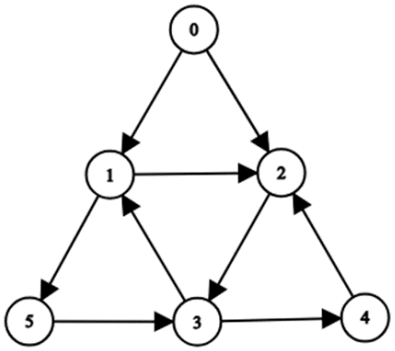


Figure 1

CO2 a) **Mention** which graph representation will let you check whether any two buildings have a bridge between them in O(1) time. **01**

b) From Figure 1

CO1 i. **Simulate** an algorithm to find the Strongly Connected Components (SCC). **04**

CO3 ii. Suppose a bridge gets blocked. **Explain** how you can efficiently calculate the updated SCCs from b(i) without rerunning the algorithm on the entire graph. **02**

CO3 c) A building is called "*Accessible*" if it's possible to start from there, visit some (or all) of the other buildings, and eventually return to that building. **03**

Propose an algorithm with a pseudocode/programmable code/step-by-step logical instructions to determine whether every building on the campus is *Accessible*. The algorithm should run in **at most O(V × (V + E))** time.

2 Captain Jack Sparrow has arrived at a treasure island, planning to steal the treasure quickly. The island has two-way pathways, each with hidden dangers. To handle these dangers, Jack brought a weapon. Facing danger in a pathway requires using some bullets. He has a map of the island showing the number of bullets needed to fight the dangers in the pathways. Bullet requirements for each pathway are not necessarily the same.

Now, answer the following questions.

a. CO1 Jack has been given the map as an Adjacency List where the locations are numbered from 1 to N. Jack is in chamber 1 and aiming to reach location N using a path that requires minimum bullets. **02**

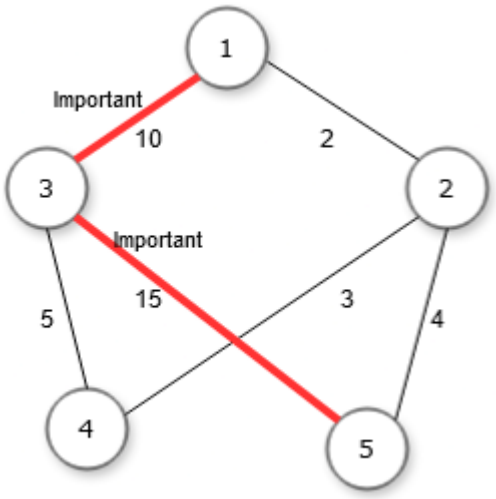
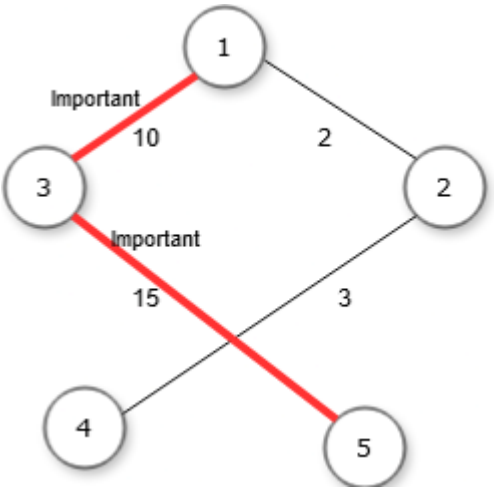
Name the algorithm you prefer to solve this problem. **Explain** your choice.

b. CO2 If some pathways contain hidden supplies of bullets that will refill his ammo by a certain amount, will your proposed algorithm in (a) work? **Explain** Briefly. **01**

c. CO2 Jack realizes that he left some of his belongings in some pathways. Hence, he wants to construct the MST such that those tunnels are <u>always</u> included no matter how dangerous they are. **04**

Formally, he has an undirected weighted graph with some edges marked as "important" and he needs to propose a modification to Kruskal's algorithm such that he can find the MST that always includes those important edges (even if it increases the total cost).

For simplicity, you can assume the important edges/tunnels do not create any cycles. Look at the example below.

Input Graph	Modified Minimum Spanning Tree
	

Present your algorithm with a programmable code/ pseudocode/ step-by-step instruction.

d. **CO3** The map of the island is too big to carry. Thus, your friend came up with an idea to use Kruskal's Algorithm to only store the Minimum Spanning Tree (MST) of the graph. He stated that "*The minimum energy needed to travel between any two chambers in the MST is the same as in the Original Map*". **03**

Do you agree with the statement? **Justify** your answer with proper examples and necessary details.

3 Consider the following scenario

You are designing a data compression system for a critical database server. You have analyzed the following character frequencies from the database logs:

Character	Frequency
A	50
B	15
C	20
D	10
E	5
F	5
G	30

Now answer the following questions:

a. **CO1** **Construct** a Huffman tree for the given characters and their frequencies. Additionally, calculate the space saved compared to using fixed 3-bit encoding for each character. Show your work step by step. **04**

b. **CO1** **Write** the Huffman Code for each character. **02**

c. **CO2** **Explain** how greedy property is used in Huffman coding and how it provides an optimal code. **02**

d. **CO2** Two friends are tasked with building the given Huffman tree. While both friends use priority queues in the process, **02**

- friend 1 inserts the newly created nodes at the end of the same frequency nodes.
- friend 2 inserts the newly created node at a random position between the same frequency nodes.

Explain which technique will provide more optimal code and why. If they are bound to provide equally optimal code, **explain** why.

4 Answer the following questions:

a. Can merge sort be written as a dynamic programming algorithm? **Explain** your answer. **02**
CO2

b. We are calculating the longest common subsequence (LCS) between two strings, $S= X_1X_2X_3X_4$ and $T = Y_1Y_2Y_3Y_4$. To do this, we fill an array C where $C_{i,j}$ represents the length of the LCS between the prefix of length i from S and the prefix of length j from T.

Some of the entries in the array C are currently masked using different symbols other than integers as shown below.

	Y_0	Y_1	Y_2	Y_3	Y_4
X_0	0	0	0	0	0
X_1	0	0	0	0	1
X_2	0	0	0	1	1
X_3	0	0	1	1	♡
X_4	0	1	♣	♠	♢

CO1
i. What does the cell $C_{0,1}$ represent? **01**
ii. What can be said about the characters X_2 and Y_3? **01**
iii. What can be said about the characters X_3 and Y_3? **01**
iv. What is the value of ♣? **01**
v. What is the value of ♠? **01**
vi. When we are filling up the i-th column of our dynamic programming table C, what columns do we need to have access to? **01**
CO3 vii. Given the observation above can we optimize our space complexity further? **02**