

Advanced Algorithms

601326

Final Exam – 2024 2

Your ID



Your Instructor Name:

Instructions for the Exam:

- Write your name and ID number on the exam and answer sheets.
- Write the number of the section that you enrolled in.
- Write the name of your instructor.
- Questions in the exam not allowed.
- Using any type of technology (mobiles, smart watches, etc.) not allowed
- Using extra papers or sheets not allowed

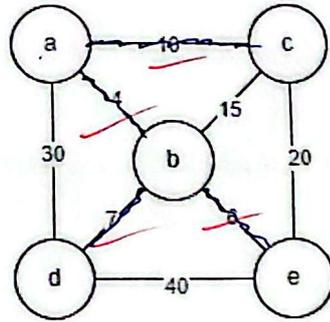
For instructor use only:

Question number	Course ILO	Program ILO	Question weight	Student mark
Q1			5	4.5
Q2			8	4
Q3			3	0
Q4	I2		4	1
Q5			5	5
Q6			4	4
Q7	I2		6	4.5
Q8			3	0
Q9			7	2
Total			45	25

This exam has 9 Questions. The total mark is 45

Question 1) Apply Prim's algorithm to the following graph to find MST showing detailed steps. (5 marks)

4.5



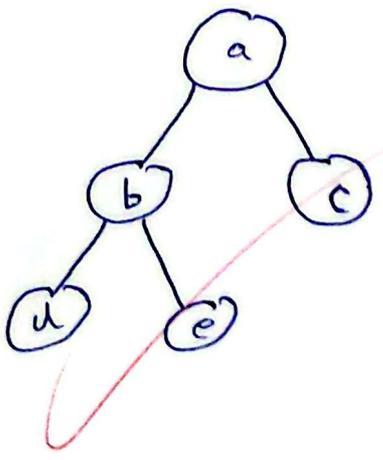
start at (a)

~~{4, 10, 30}~~ we take edge (4) {4}

~~{4, 7, 10, 15, 30}~~ we take edge (6) {4, 6}

~~{4, 7, 10, 15, 20, 30, 40}~~ {4, 6, 7} {4, 6, 7, 10}

?!
!!



~~Handwritten scribbles and crossed-out text at the top of the page.~~

it will start looping because there is a shorter path to 1 from zero

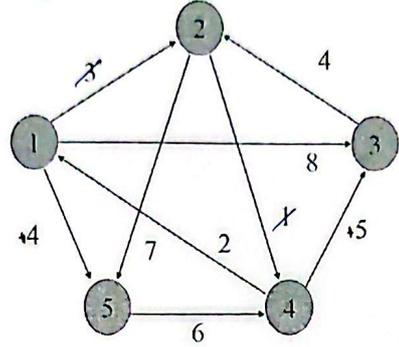
0.5

4

Question 2)

(8 marks)

- a. Calculate the shortest path from vertex 1 to each vertex of the below graph using Dijkstra's algorithm
- b. Give example shows that Dijkstra's algorithm may not work for a weighted connected graph with negative weights.
- c. Discuss the time complexity of the algorithm and the factors affecting it.



~~Handwritten scribbles and crossed-out text for part a.~~

Handwritten Dijkstra's algorithm steps showing priority queues for each vertex:

- 1 { - , 0 }
- 2 { 1, 3 }
- 3 { 1, 8 }
- 4 { - , 0 }
- 5 { 1, 4 }
- 4 { 2, 4 }
- 5 { 1, 4 }
- 3 { 4, 9 }
- 5 { 1, 4 }
- 3 { 4, 9 }

3.75

$E \log V$

$V \log E$ because it goes through all the vertex but not all edges

~~Handwritten scribbles and crossed-out text for part c.~~

Question 3) Based on your understanding of Floyd's algorithm (for weighted graphs), write a recursive relation for finding R^{k}_{ij} . (3 marks)

0

?

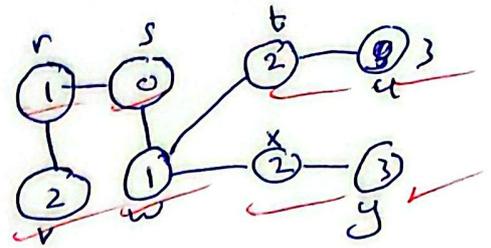
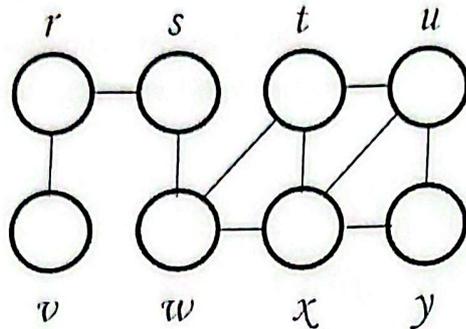
Question 4) Fill in the bellow table best and worst cases of the following algorithms: (4 marks)

1

	Insertion Sort	Quick Sort	Heap Sort	Merge Sort
Best Case	$\log n$	X	X	$\log n$
Worst Case	$n \log n$	$\log n$	$\log n$	$\log n$

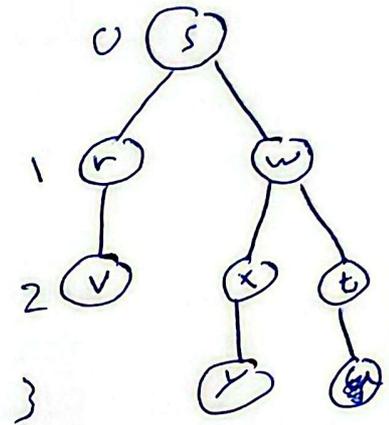
5

Question 5) Given the below graph, apply the BFS algorithm to find distance from source vertex s to all other vertices and find BFS tree (5 marks)



$\{\}$
 $\{s\} \rightarrow s = 0$
 $\{r, w\} \rightarrow r = 1$
 $\{w, v\} \rightarrow w = 1$

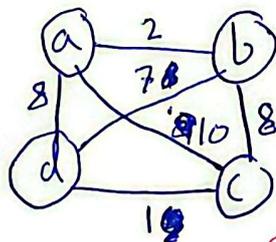
$\{v, t, x\} \rightarrow v = 2$
 $\{t, x\} \rightarrow t = 2$
 $\{x, y\} \rightarrow x = 2$
 $\{u, y\} \rightarrow u = 3$
 $\{y\} \rightarrow y = 3$



Question 6) Why can NOT the Traveling Salesman Problem (TSP) be solved efficiently using a greedy algorithm? Provide an example to support your answer. (4 marks)

because greedy algorithms gets the local optimal and (TSP) is a optimization problem so we want a global optimal

4



in a greedy algo we take the shortest path we see so we take $a \rightarrow b$ and it gives $b \rightarrow d$ as a value of 20
 $d \rightarrow c$
 $c \rightarrow a$

but there is a better way $a \rightarrow b$ and it gives $b \rightarrow c$ as a value of 19
 $c \rightarrow d$
 $d \rightarrow a$

Question 7) Discuss the difference between the following:

(6 marks)

4.5

- 1- P and NP Problems
- 2- Decision and Optimization problems
- 3- Branch & Bound and Brute Force techniques

- 1- P can be solved in polynomial time and you can check if it's the true answer really ^{easily} and NP can't be solved in polynomial time so ~~you~~ you can't check the answer easily 1.5
- 2- Decision problems are problems that have a true or false answer but optimization problems are problems that you have to check for the best answer 2
- 3- Brute force goes through all possible answers and takes the best so it has a very bad time complexity but Branch and Bound ?! 1

Question 8)

(3 marks)

Write the recursive formulation for the 0/1 Knapsack Problem for a Dynamic Programming solution

0

?

Question 9)

(7 marks)

Design greedy algorithm for solving the fractional knapsack problem:
given weights and values of n items, put these items into a knapsack of capacity W to get the maximum total value in the knapsack (note: you can take fractions of an item).

2

$t = \text{items } \{ \overset{\text{weight}}{w}, \overset{\text{value}}{v} \}$
 $W = \text{max weight}$
for all items⁽ⁱ⁾
for every w
val value $f = \frac{t\{v\}}{t\{w\}}$ $t\{v\}/t\{w\}$
for all items-1⁽ⁱ⁾
small $= \min\{f_i, f_{i+1}\}$
add δ small to a list

1
{
{
{

Good Luck