

Advanced Algorithms  
601326  
Final Exam – 2024 2



- 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

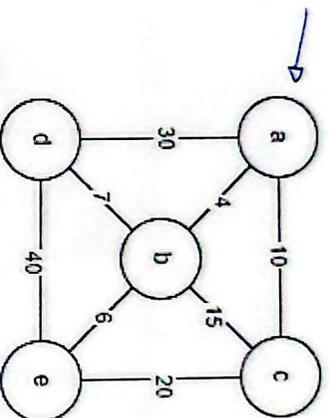
For instructor use only:

Question number	Course ILO	Program ILO	Question weight	Student mark
Q1			5	4.5
Q2			8	6
Q3			3	1.25
Q4	I2		4	3.5
Q5			5	2.5
Q6			4	2
Q7	I2		6	6
Q8			3	1.5
Q9			7	2.5
Total			45	29.75

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

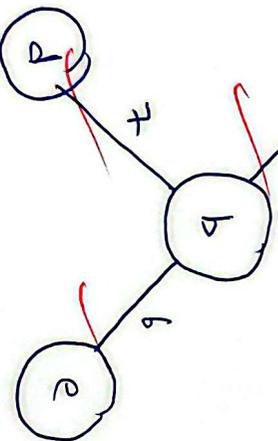
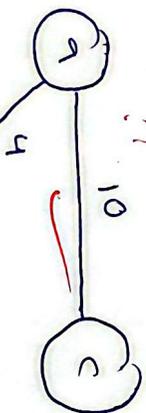


$$h = \{4, 10, 30\}$$

$$h = \{7, 10, 15, 30\}$$

$$h = \{10, 15, 30, 40\}$$

~~h = {10, 15, 30, 40}~~



$$\text{Cost} = 27$$

Question 2)

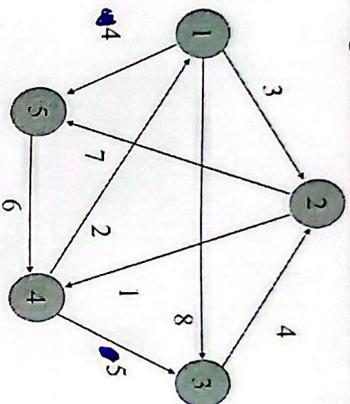
(8 marks)

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.

6



a)

$1(-, 0)$   
 $2(1, 3)$   
 $3(1, 8)$   
 $4(2, \infty)$   
 $5(1, 4)$

$2(1, 3)$   
 $3(1, 8)$   
 $4(2, \infty)$   
 $5(2, 10)$

$3(1, 8)$   
 $4(2, 4)$   
 $5(2, 10)$

$4(3, \infty)$   
 $5(3, \infty)$

$5(4, \infty)$   
 $3(4, 9)$

$3(5, \infty)$

Final shortest paths from vertex 1:  
 1:  $(-, 0)$   
 2:  $(1, 3)$   
 3:  $(1, 8)$   
 4:  $(2, 4)$   
 5:  $(1, 4)$

b)

$A(-, 0)$        $B(A, 2)$        $C(A, 3)$   
 $* B(A, 2)$        $C(B, 0)$   
 $C(B, 0)$

2

The shortest path for B is -2 from C  
 so negative weights give wrong paths

c)  $\rightarrow$  0.5

c) Time complexity is  $O(E \log V)$

The factors that affect ~~it~~ a number of  $V$ .

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

~~Recursive relation for finding  $R_{ij}^k$~~

$$R_{ij} = \min(R_{ij}, R_{ik} + R_{kj}) \text{ , } i, j > 0$$

1.25

?

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

3.5

	Insertion Sort	Quick Sort	Heap Sort	Merge Sort
Best Case	<del><math>O(n)</math></del>	<del><math>O(n \log n)</math></del>	<del><math>O(n \log n)</math></del>	<del><math>O(n^2)</math></del>
Worst Case	<del><math>O(n^2)</math></del>	<del><math>O(n^2)</math></del>	<del><math>O(n \log n)</math></del>	<del><math>O(n^2)</math></del>

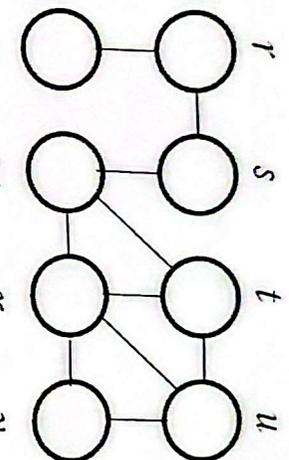
if we used  
 $\downarrow$   
 median of 3  
 we can get  
 $O(n \log n)$  even  
 in worst case

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)

Q.5

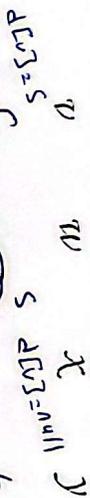
$h = \{s\}$



missing distance !!

$h = \{s, r, v\}$

$h = \{s, r, v, w\}$

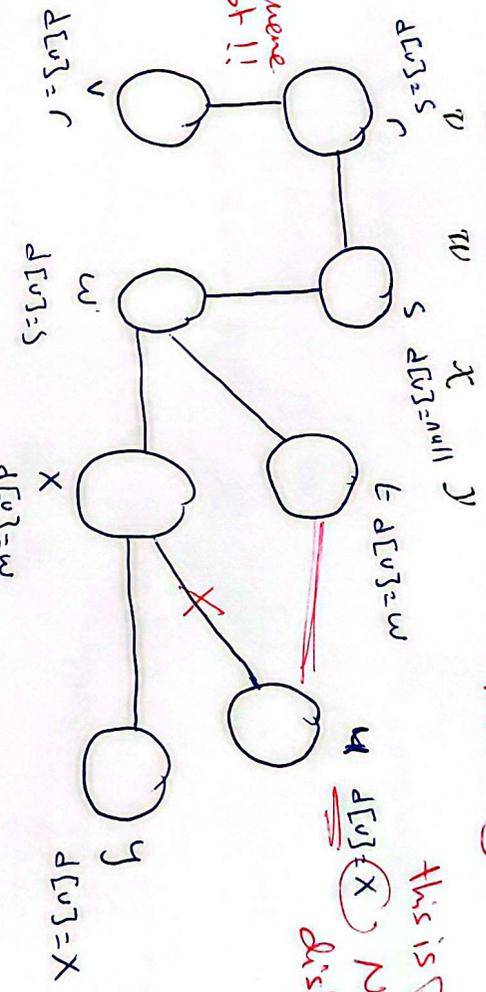


This is Parent NOT distance

$h = \{s, r, v, w\}$

You should degenerate + First !!

$h = \{s, r, v, w, y\}$



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)

greedy algorithms looking for the best solution overall

2

while TSP can be complexed and may not reach a solution as quick as possible.

Question 7) Discuss the difference between the following:

(6 marks)

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



1) P is Polynomial time taken for quickly solve a problem

NP is non-deterministic polynomial for quickly checked the solutions of the problem.

2) Decision is Yes/No Problems, ex: Hamiltonian algorithm, easier than optimization.

Optimization is seeking for optimal solution (best solution), can be harder than decision, ex: 0/1 Knapsack problem

3) Branch and Bound is Branch for going through all vertex. Bound to remove unnecessary solutions.

Brute Force is checking every solution worse than Branch and Bound.

Question 8)

(3 marks)

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

~~$V(i, w) = 0$~~ ,  $i, w = 0$

$V(1, 5)$

$V(i, w) = \max(V(i-1, w), V(i, w) + V(i-1, w-w_i))$  if  $w > w$

missing case  $V(i-1, w)$

## 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).

2.5

For  $i=0$  to  $W$  :

$D_{ni} = 0$

// first row and column set to 0.

For  $j=0$  to  $n$  :

$D_{jw} = 0$

For  $i=1$  to  $n$  :

For  $j=1$  to  $w$  :

$v(i,j) = \max(v(i,j-1), v(i,j) + v(i,j-j_i))$

2.5

Good Luck