BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Semester Final Exam
Duration: 1 Hour 40 Minutes

Semester: Spring 2023
Full Marks: 40

CSE 221: Algorithms

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

Name:	ID:	Section:

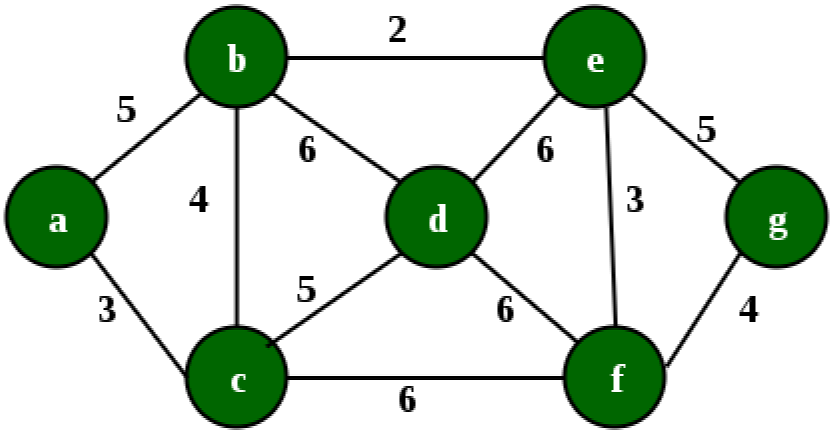
1 **a. CO2**

You work as an engineer for the Roads and Highways department in a district. The district has 7 thanas that are represented by the 7 vertices in the following graph. The edges between the vertices represent the roads that connect one thana with another and the weight of an edge represents the length of the road.

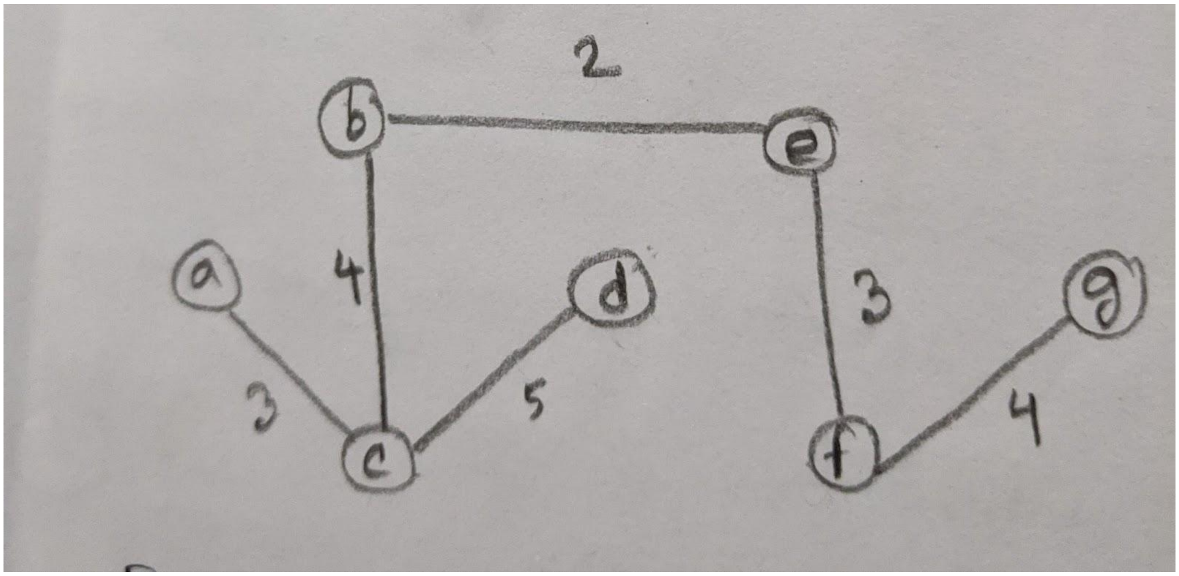
A recent flood has totally damaged these roads and immediate repair work is needed. The cost to repair a road is proportionate to its length. However, your department does not have the budget to repair all the roads so you need to repair a subset of the roads. Which algorithm should you use here to find out the roads that need to be repaired to keep all the thanas connected to one another with the **minimum** possible cost?

Show the step by step simulation of the algorithm. You may pick any algorithm of your preference that you think will solve the problem.

07



Solution:



The length of the the MST is 2+3+4+3+4+5=21

b.
CO3

Problem Y is NP-complete. If A professor takes an instance of Y, converts it into an instance of the shortest path problem in 2^n time, and claims that Y is in P now, is the claim valid?
Explain your answer.

Solution: The claim is not valid. This is an example of reduction and in reduction, the time required to convert an instance into another one should be polynomial but 2^n is exponential.

02

c.
CO3

Rank the following three approximation algorithms from the worst to the best:
2-approximation algorithm, 1.5-approximation algorithm, n-approximation algorithm

Solution: worst is n-approximation, next 2-approximation, best is 1.5-approximation

01

2

Two infamous thieves, Denver and Nairobi, planned to rob the famous Louvre Museum. Before the scene, they both agreed on the fact that none of them will break any item as all the items in the museum are too precious, and taking a fraction of any item won't sell on the black market. If it fits in the bag as a whole, they will take it, otherwise, leave it as it is.

Both of them arrived at the Royal Treasury with an empty knapsack weighing a total of **7 kg** each. Even though both thieves are experts in their fields, they take slightly different approaches. Denver believes he will use a Dynamic Programming Approach to rob the items in the most efficient manner possible. Nairobi, on the other hand, believes that if she chooses a Greedy Approach, she will make the most money.

The objects in the Royal Treasury Museum are listed below.

Objects	Diamond	Jewelry	Sculpture	Painting	Gold Crest
Profit ($)	3	4	12	9	12
Weight (kg)	1	2	8	4	5

CO2 **05**

CO2 a) **Simulate** your dynamic programming algorithm to find the maximum profit Denver can make. **02**

CO3 b) From your memory/dp table, **find** out which items would he take to make this amount of profit? **03**

c) Does Nairobi’s belief remain valid after the robbery? **Prove** it.

Solutions

a)

Knapsack 0-1

Resultant Table

Item	Profit	Weight	0	1	2	3	4	5	6	7
0 (Diamond)	3	1	0	3	3	3	3	3	3	3
1 (Jewelry)	4	2	0	3	4	7	7	7	7	7
2 (Sculpture)	12	8	0	3	4	7	7	7	7	7
3 (Painting)	9	4	0	3	4	7	9	12	13	16
4 (Gold Crest)	12	5	0	3	4	7	9	12	15	16

Resultant Profit

16

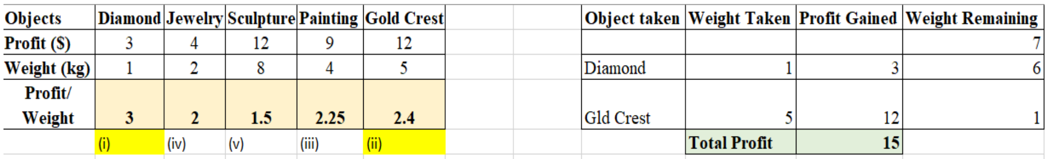
Profit

3,4,9

Item No.

0,1,3

b)



Objects	Diamond	Jewelry	Sculpture	Painting	Gold Crest
Profit ($)	3	4	12	9	12
Weight (kg)	1	2	8	4	5
Profit/ Weight	3	2	1.5	2.25	2.4
	(i)	(iv)	(v)	(iii)	(ii)

Object taken	Weight Taken	Profit Gained	Weight Remaining
			7
Diamond	1	3	6
Gld Crest	5	12	1
	Total Profit	15	

c)

Since Nairobi getting less profit than Denver ($16), so her belief doesn't remain valid.

3 Following are the codes generated from a text for a Huffman tree construction.

H - 1000 *o* - 1001 *<space>* - 1010	*u* - 000 *d* - 001 *n* - 010	*e* - 011 *l* - 110 *t* - 111

S - 1011		

You are also given the following information:

- The frequency of each leaf node except *e*, *l*, and *t* is 1.
- The left and right child nodes of the root have frequencies 5 and 8 respectively.

Now answer the following questions.

CO2

a) Suppose in a Huffman tree, the distances from the root to the pair of leaves denoting the letters *k* and *b* are 5 and 2 respectively, which letter between them is more frequent in the original text? Just **mention** the letter. 01

b) **Draw** the Huffman tree from the given coding table above. 03

c) Continuing on Q(b), what are the frequencies of *l*, and *t* in the original text? Just **mention** the frequencies. 02

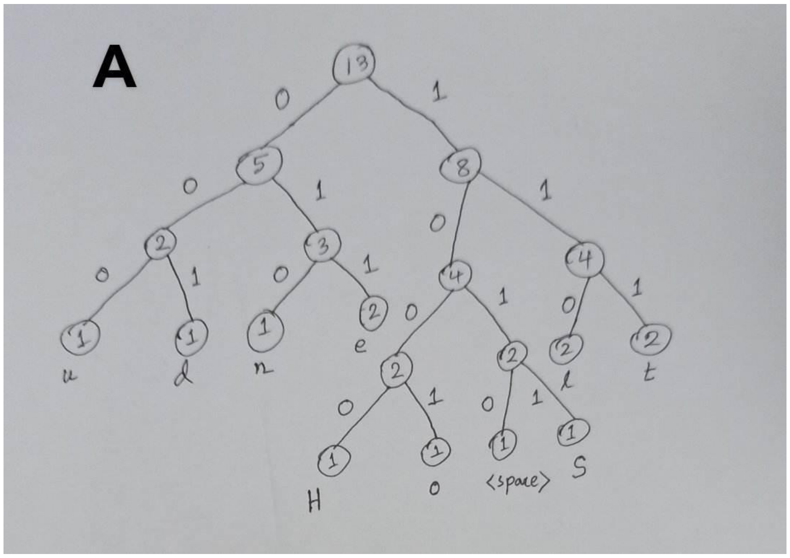
Your lazy friend, Tom, used a very simple encoding method for compressing a text file. The text file contains the string : *Sudden Hello*. He used a constant number of bits for encoding each of the ***n*** distinct characters in this text. The constant is $ceil(log_2 n)$ in this case. On the other hand, you have compressed the text file using your own huffman tree from Q(b).

d) **Mention** how many bits each of the characters in Tom's encoding scheme contain. 01

e) **Compare** between the number of bits needed to decode the above-mentioned string using your friend's scheme and your Huffman tree from Q(b). 03

Solutions (question 3)

a) b



b) Tree:

c) 2,2

d) 4

e) Huffman encoding. Because, Tom's encoding needs 12 x 4 = 48. The Huffman encoding scheme needs 40 bits (4+3+3+3+3+3+4+4+3+3+3+4 = 40).

Answer only one of 4, 4(or)

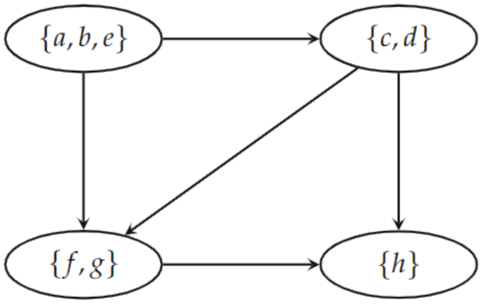
4 Your friend is trying to find the strongly connected components of a directed graph G. Recall that this involves first running depth-first search on G, and then running depth-first search on the reverse graph, G^R, using the decreasing order of finish time. Your friend has done this and found the following table. Here,a, b, c, d, e, f , g, h, i are the vertices of G.

	a	b	c	d	e	f	g	h	i
`finish`(G)	6	4	8	18	5	14	17	16	15
`finish`(G^R)	18	17	12	10	16	6	8	9	4

Now answer the following questions.

CO2

a) **Find** a vertex in G that is guaranteed to be contained in a source connected component. Recall that a source connected component in an SCC DAG is a node that has indegree zero. For example, in the following SCC DAG, $\{a, b, e\}$ is a source connected component. **01**



This image is just an example to show how SCC can convert a graph into DAG, and the source connected component

b) Let x be your answer to (a). **Find** all the vertices in the strongly connected component that contains x. **05**

c) If the vertices you found in (b) are deleted from G, which set of vertices is guaranteed to be a source connected component of the resulting graph? **Write** the list of those vertices. **01**

d) How many strongly connected components does G have? Just **write** the answer. **01**

e) **Draw** the DAG of the strongly connected components of G. There is more than one correct answer, anyone of those will be accepted. **02**

Solutions

Solution:

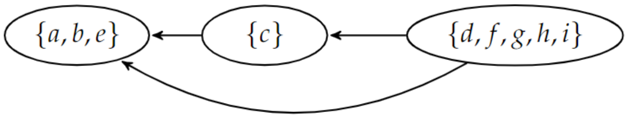
(a) d if only using information from the first row of the table. If using both rows of the table, then any vertex from (b) works.

(b) $\{d, f, g, h, i\}$.

(c) $\{c\}$.

(d) three.

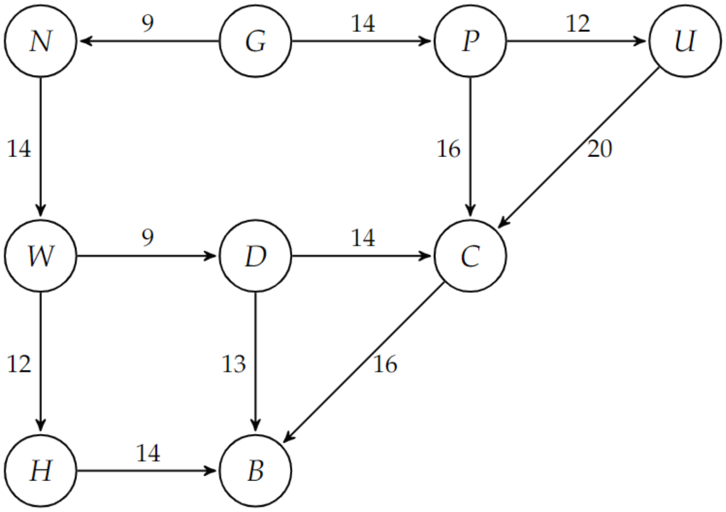
(e) The correct answer is any graph obtained by deleting zero or more edges from the following.



Question 4 (or) is on the next page.

---

4 **or** Consider the network of roads connecting a set of cities given by the graph below.



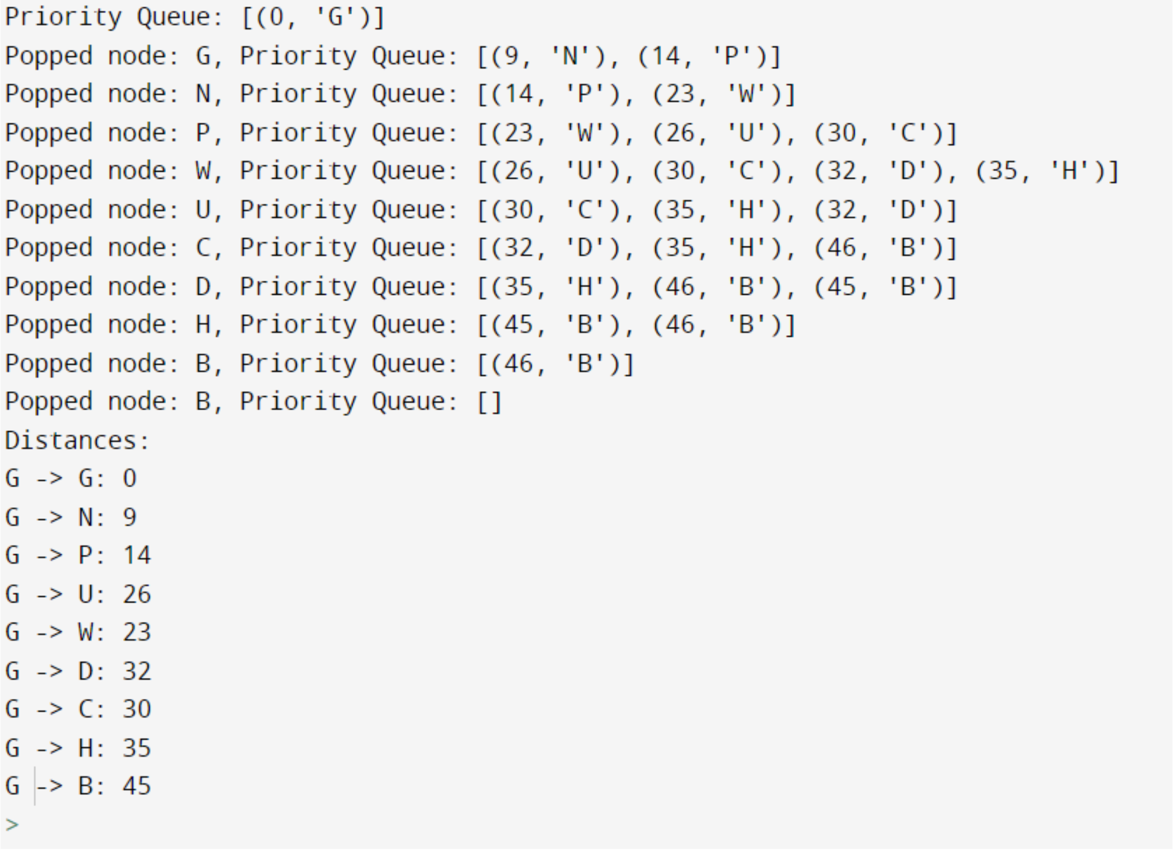
CO2

a) **Simulate** Dijkstra's algorithm to find the weight of the shortest path from G to B. **08**

b) There is a proposal to add one new road to this network, and there is a list of five pairs of cities between which the new road can be built. Each such potential road has an associated length. As a designer for the public works department, you are asked to determine the road whose addition to the existing network would result in the maximum decrease in the distance from G to B. One way to do this would be to add each of the five edges separately, run Dijkstra's algorithm five different times, and then compare the results. But can you do **02** this by invoking Dijkstra’s algorithm only two times (and possibly doing a linear amount of extra work)?

Present your idea with a pseudo-code/flowchart/step by step instruction/algorithm explanation.

Solutions



```
Priority Queue: [(0, 'G')]
Popped node: G, Priority Queue: [(9, 'N'), (14, 'P')]
Popped node: N, Priority Queue: [(14, 'P'), (23, 'W')]
Popped node: P, Priority Queue: [(23, 'W'), (26, 'U'), (30, 'C')]
Popped node: W, Priority Queue: [(26, 'U'), (30, 'C'), (32, 'D'), (35, 'H')]
Popped node: U, Priority Queue: [(30, 'C'), (35, 'H'), (32, 'D')]
Popped node: C, Priority Queue: [(32, 'D'), (35, 'H'), (46, 'B')]
Popped node: D, Priority Queue: [(35, 'H'), (46, 'B'), (45, 'B')]
Popped node: H, Priority Queue: [(45, 'B'), (46, 'B')]
Popped node: B, Priority Queue: [(46, 'B')]
Popped node: B, Priority Queue: []
Distances:
G -> G: 0
G -> N: 9
G -> P: 14
G -> U: 26
G -> W: 23
G -> D: 32
G -> C: 30
G -> H: 35
G -> B: 45
>
```

(b) (sketch) - Run Dijkstra’s algorithm from G on the original graph. Let d_G(v) be the distance values.

- **Run Dijkstra’s algorithm from B on the reverse graph. Let d_B(v) the distance values.**
- **Find the edge (u, v) that minimizes the quantity is d_G(u)+w(u, v) + d_B(v).**

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Mid Semester Exam
Duration: 1 Hour 40 Minutes

Semester: Spring 2023
Full Marks: 40

CSE 221: Algorithms

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

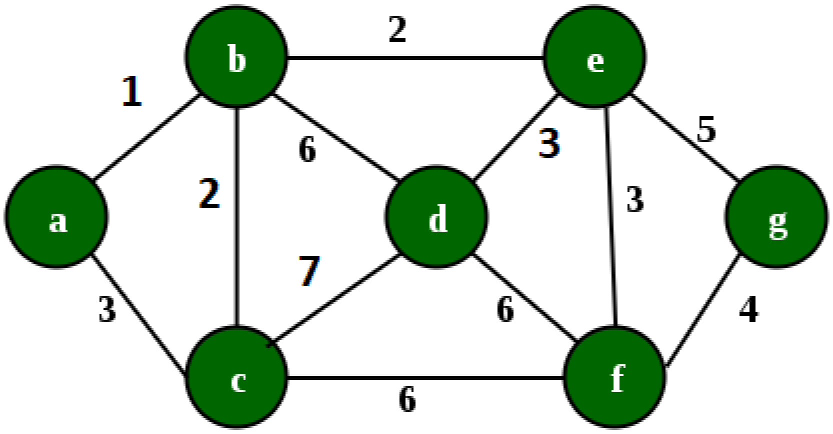
Name:	ID:	Section:

1 **a.** **CO2** **07**

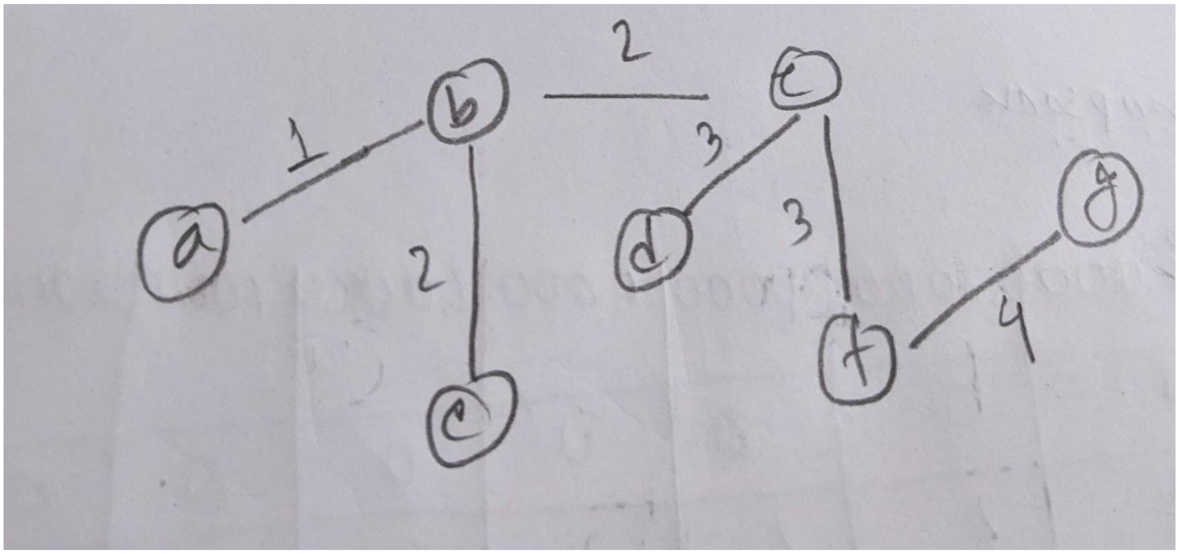
You work as an engineer for the Roads and Highways department in a district. The district has 7 thanas that are represented by the 7 vertices in the following graph. The edges between the vertices represent the roads that connect one thana with another and the weight of an edge represents the length of the road.

A recent flood has totally damaged these roads and immediate repair work is needed. The cost to repair a road is proportionate to its length. However, your department does not have the budget to repair all the roads so you need to repair a subset of the roads. Which algorithm should you use here to find out the roads that need to be repaired to keep all the thanas connected to one another with the **minimum** possible cost?

Show the step by step simulation of the algorithm. You may pick any algorithm of your preference that you think will solve the problem.



Solution:

		 **The length of the MST is 1+2+2+3+3+4=15**	
	b. **CO3**	Problem Y is NP-complete. If A professor takes an instance of Y, converts it into an instance of the shortest path problem in n^3 time, and claims that Y is in P now, is the claim valid? **Explain** your answer. Solution: The claim is valid. This is an example of reduction and in reduction, the time required to convert an instance into another one should be polynomial. The time complexity n^3 is polynomial.	**02**
	c. **CO3**	**Rank** the following three approximation algorithms from the worst to the best: n-approximation algorithm, 2-approximation algorithm, 5-approximation algorithm Solution: Worst is n-approximation, next is 5-approximation, best is 2-approximation	**01**

2

CO2
CO3

07
03

You are Monco, a video game character, you have just stepped into level 2. This game requires you to acquire necessity items to sustain through a level and this is done by trading reward points from the previous level. From level 1 you have gained 9,000 points. Below is a list of items you can get by trading your reward points. Each item has certain properties but be wise to pick the ones which will maximize your sustainability by maximizing mission impact.

	Gold Bar	Pain Killer	Waterproof Boots	Sun glasses	Energy Bar	Night Vision
Trade Price (points)	8000	2000	2000	2000	1000	3000
Mission Impact	1	5	3	2	2	4

a) **Simulate** your chosen algorithm on the below items to find your picked items.
b) If you were to apply brute force, what would be the time complexity and why? **Explain.** Assume that points you have gained are denoted as P, and the number of items is N.

Solutions

b) Need to find all the combination of the items and select that combination whose total price is less or equal to 9000 and impact is maximum. O(2^N)

Resultant Table

Item	Profit	Weight	0	1	2	3	4	5	6	7	8	9
0 (Gold Bar)	1	8	0	0	0	0	0	0	0	0	1	1
1 (Pain KIller)	5	2	0	0	5	5	5	5	5	5	5	5
2 (Waterproof Boots)	3	2	0	0	5	5	8	8	8	8	8	8
3 (Sunglasses)	2	2	0	0	5	5	8	8	10	10	10	10
4 (Energy Bar)	2	1	0	2	5	7	8	10	10	12	12	12
5 (Night Vision)	4	3	0	2	5	7	8	10	11	12	14	14

Resultant Profit

14

Profit

5,3,2,4

Item No.

1,2,3,5

3

Following are the codes generated from a text for a Huffman tree construction.

J - 1000	*n* - 000	*l* - 011
o - 1001	*h* - 001	*e* - 110
<space> - 1010	*m* - 010	*c* - 111
W - 1011		

You are also given the following information:

- The frequency of each leaf node except *e*, *l*, and *c* is 1.
- The left and right child nodes of the root have frequencies 5 and 8 respectively.

Now answer the following questions.

CO2

a) Suppose in a Huffman tree, the distances from the root to the pair of leaves denoting the letters *m* and *e* are 3 and 5 respectively, which letter between them is more frequent in the original text? Just **mention** the letter. **01**

b) **Draw** the Huffman tree from the given coding table above. **03**

c) Continuing on Q(b), what are the frequencies of *e*, and *c* in the original text? Just **mention** the frequencies. **02**

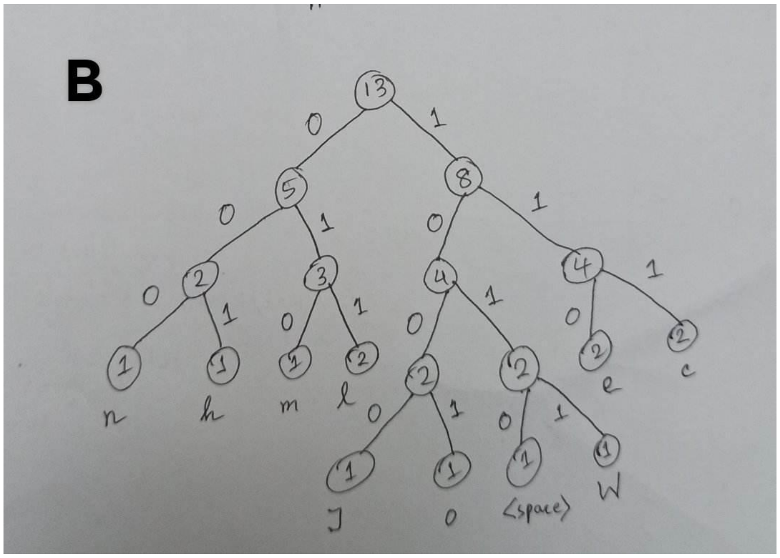
Your lazy friend, Tom, used a very simple encoding method for compressing a text file. The text file contains the string : *Welcome John*. He used a constant number of bits for encoding each of the ***n*** distinct characters in this text. The constant is $ceil(log_2 n)$ in this case. On the other hand, you have compressed the text file using your own huffman tree from Q(b).

d) **Mention** how many bits each of the characters in Tom's encoding scheme contain. **01**

e) **Compare** between the number of bits needed to decode the above-mentioned string using your friend's scheme and your Huffman tree from Q(b). **03**

Solutions (question 3):

a) m.



b) Tree:

c) 2,2

d) 4

e) Huffman encoding. Because, Tom's encoding needs 12 x 4 = 48. The Huffman encoding scheme needs 41 bits (4+3+3+3+4+3+3+4+4+4+3+3 = 41).

Answer only one of 4, 4(or)

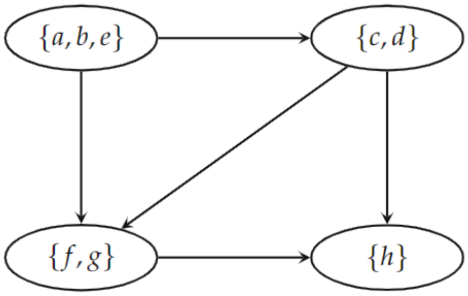
4 Your friend is trying to find the strongly connected components of a directed graph G. Recall that this involves first running depth-first search on G, and then running depth-first search on the reverse graph, G^R, using the decreasing order of finish time. Your friend has done this and found the following table. Here, a, b, c, d, e, f, g, h, i are the vertices of G.

	a	b	c	d	e	f	g	h	i
`finish(G)`	7	8	9	14	6	18	13	17	16
`finish(`G^R`)`	16	18	12	6	17	10	7	8	9

Now answer the following questions.

CO2

a) **Find** a vertex in G that is guaranteed to be contained in a source connected component. Recall that a source connected component in an SCC DAG is a node that has indegree zero. For example, in the following SCC DAG, $\{a, b, e\}$ is a source connected component. **01**



This image is just an example to show how SCC can convert a graph into DAG, and the source connected component

b) Let x be your answer to (a). **Find** all the vertices in the strongly connected component that contains x. **05**

c) If the vertices you found in (b) are deleted from G, which set of vertices is guaranteed to be a source connected component of the resulting graph? **Write** the list of those vertices. **01**

d) How many strongly connected components does G have? Just **write** the answer. **01**

e) **Draw** the DAG of the strongly connected components of G. There is more than one correct answer, anyone of those will be accepted. **02**

Solutions

Solution:

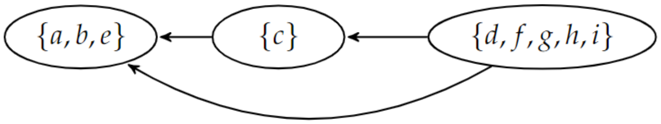
(a) f if only using information from the first row of the table. If using both rows of the table, then any vertex from (b) works.

(b) $\{d, f, g, h, i\}$.

(c) $\{c\}$.

(d) three.

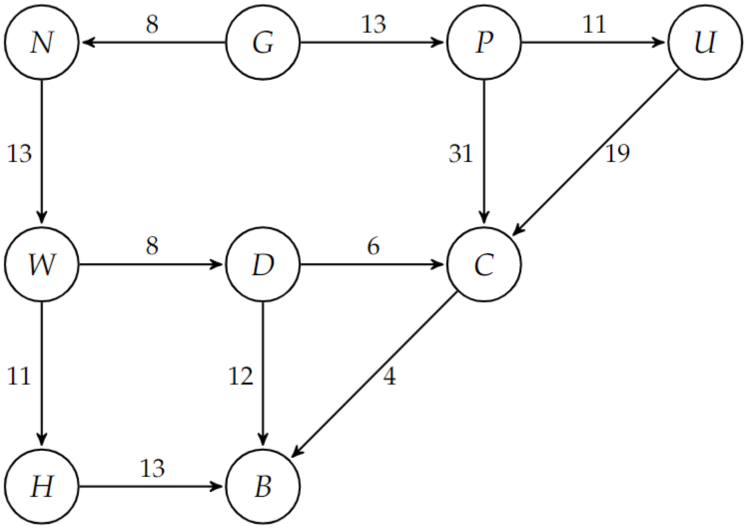
(e) The correct answer is any graph obtained by deleting zero or more edges from the following.



Question 4 (or) is on the next page.

4 **or**

Consider the network of roads connecting a set of cities given by the graph below.



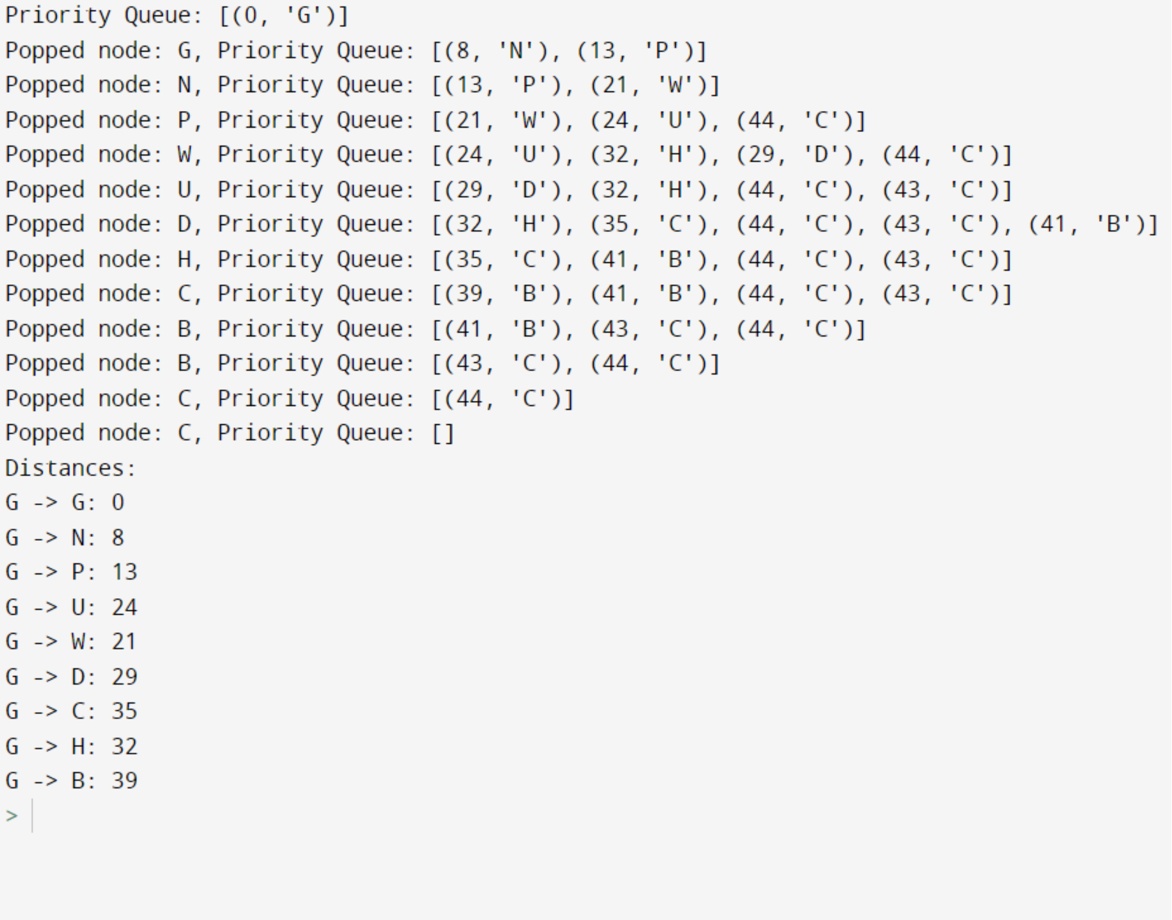
CO2

a) **Simulate** Dijkstra's algorithm to find the weight of the shortest path from G to B. **08**

b) There is a proposal to add one new road to this network, and there is a list of five pairs of cities between which the new road can be built. Each such potential road has an associated length. As a designer for the public works department, you are asked to determine the road whose addition to the existing network would result in the maximum decrease in the distance from G to B. One way to do this would be to add each of the five edges separately, run Dijkstra's algorithm five different times, and then compare the results. But can you do this by invoking Dijkstra's algorithm only two times (and possibly doing a linear amount of extra work)? **02**

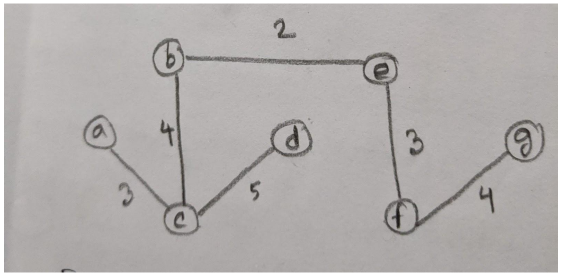
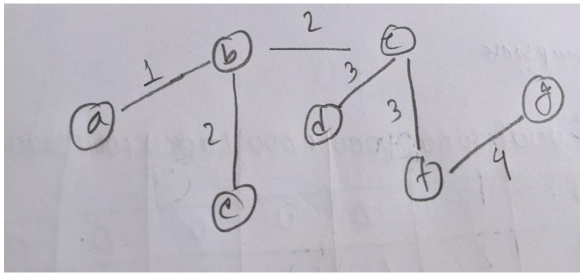
Present your idea with a pseudo-code/flowchart/step by step instruction/algorithm explanation.

Solutions



```
Priority Queue: [(0, 'G')]
Popped node: G, Priority Queue: [(8, 'N'), (13, 'P')]
Popped node: N, Priority Queue: [(13, 'P'), (21, 'W')]
Popped node: P, Priority Queue: [(21, 'W'), (24, 'U'), (44, 'C')]
Popped node: W, Priority Queue: [(24, 'U'), (32, 'H'), (29, 'D'), (44, 'C')]
Popped node: U, Priority Queue: [(29, 'D'), (32, 'H'), (44, 'C'), (43, 'C')]
Popped node: D, Priority Queue: [(32, 'H'), (35, 'C'), (44, 'C'), (43, 'C'), (41, 'B')]
Popped node: H, Priority Queue: [(35, 'C'), (41, 'B'), (44, 'C'), (43, 'C')]
Popped node: C, Priority Queue: [(39, 'B'), (41, 'B'), (44, 'C'), (43, 'C')]
Popped node: B, Priority Queue: [(41, 'B'), (43, 'C'), (44, 'C')]
Popped node: B, Priority Queue: [(43, 'C'), (44, 'C')]
Popped node: C, Priority Queue: [(44, 'C')]
Popped node: C, Priority Queue: []
Distances:
G -> G: 0
G -> N: 8
G -> P: 13
G -> U: 24
G -> W: 21
G -> D: 29
G -> C: 35
G -> H: 32
G -> B: 39
>
```

(b) check set A

1a-7	 **2+3+4+3+4+5=21**	1a-7	 **1+2+2+3+3+4=15**
1b-2	not valid, convert should be polynomial but 2^n is exponential.	1b-2	valid
1c-1	worst is n-approximation, next 2-approximation, best is 1.5-approximation	1c-1	Worst is n-approximation, next is 5-approximation, best is 2-approximation
2a-5	3+4+9=16	2a-7	5+3+2+4=14, item 1,2,3,5 painkiller, boots, sunglass, night vision
2b-2	items: 0,1,3 diamond, jewelry, painting or jewelry, gold crest	2b-3	O(2^N)
2c-3	Not valid		
3a-1	b **3b-3** tree draw	3a-1	m **3b-3** tree draw
3c-2	2,2	3c-2	2,2
3d-1	4	3d-1	4
3e-3	Tom's encoding needs 12 x 4 = 48. The Huffman needs 40 bits	3e-3	Tom's encoding needs 12 x 4 = 48. The Huffman needs 41 bits
4a-1	D or any from D,F,G,H,I **4b-5** D,F,G,H,I **4c-1** C **4d-1** 3	4a-1	F or any from D,F,G,H,I **4b-5** D,F,G,H,I **4c-1** C **4d-1** 3
4e-2	(a,b,e) (c) (d,f,g,h,i)	4e-2	(a,b,e) (c) (d,f,g,h,i)
4oa-8	g | n | p | u | w | d | c | h | b	4oa-8	g | n | p | u | w | d | c | h | b

<table>
<tr>
<td></td>
<td><table><tr><td>0</td><td>9</td><td>14</td><td>26</td><td>23</td><td>32</td><td>30</td><td>35</td><td>45</td></tr></table></td>
<td></td>
<td><table><tr><td>0</td><td>8</td><td>13</td><td>24</td><td>21</td><td>29</td><td>35</td><td>32</td><td>39</td></tr></table></td>
</tr>
<tr>
<td>4ob
-2</td>
<td>One from G, one from B on reverse graph, find min edge (u,v) for G-u-v-B</td>
<td>4ob
-2</td>
<td>One from G, one from B on reverse graph, find min edge (u,v) for G-u-v-B</td>
</tr>
</table>