A

BRAC UNIVERSITY
Department of Computer Science and Engineering

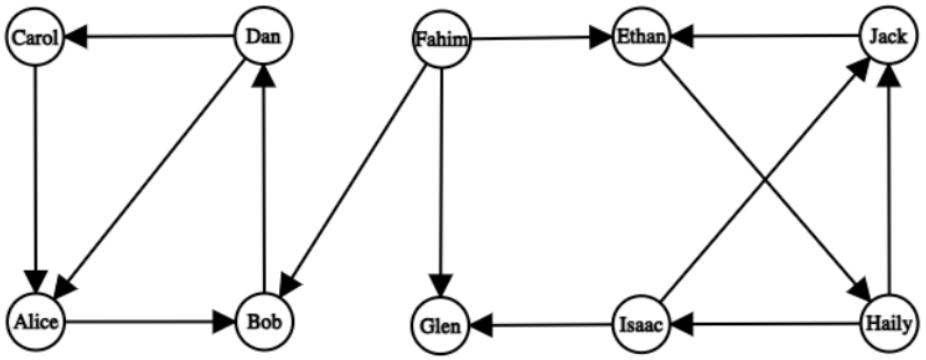
Examination: Final Exam | Semester: Summer 2023
Duration: 1 Hour 40 Minutes | Full Marks: 40

CSE 221: Algorithms

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

Name:	ID:	Section:

1. Among a group of 10 friends, it's not feasible for each individual to visit the houses of all the others. With an upcoming tournament in mind, the friends need to establish groups. The criterion for forming these groups is that a friend must be capable of visiting the houses of every other member in their group. The task involves calculating the count of viable groups following this criterion. A group has to consist of a minimum of two people.



a) [CO1] The end result will showcase the list of groups and the members. **Name** the algorithm tailored for this situation. **01**

b) [CO1] **Show** a simulation of your chosen algorithm using the graph above. **Write** the number of groups that can be formed and the members of each group. **05**

c) [CO1] Those who are incapable of forming a group will be identified as individuals ineligible for tournament participation. **Identify** those people if there are any. **01**

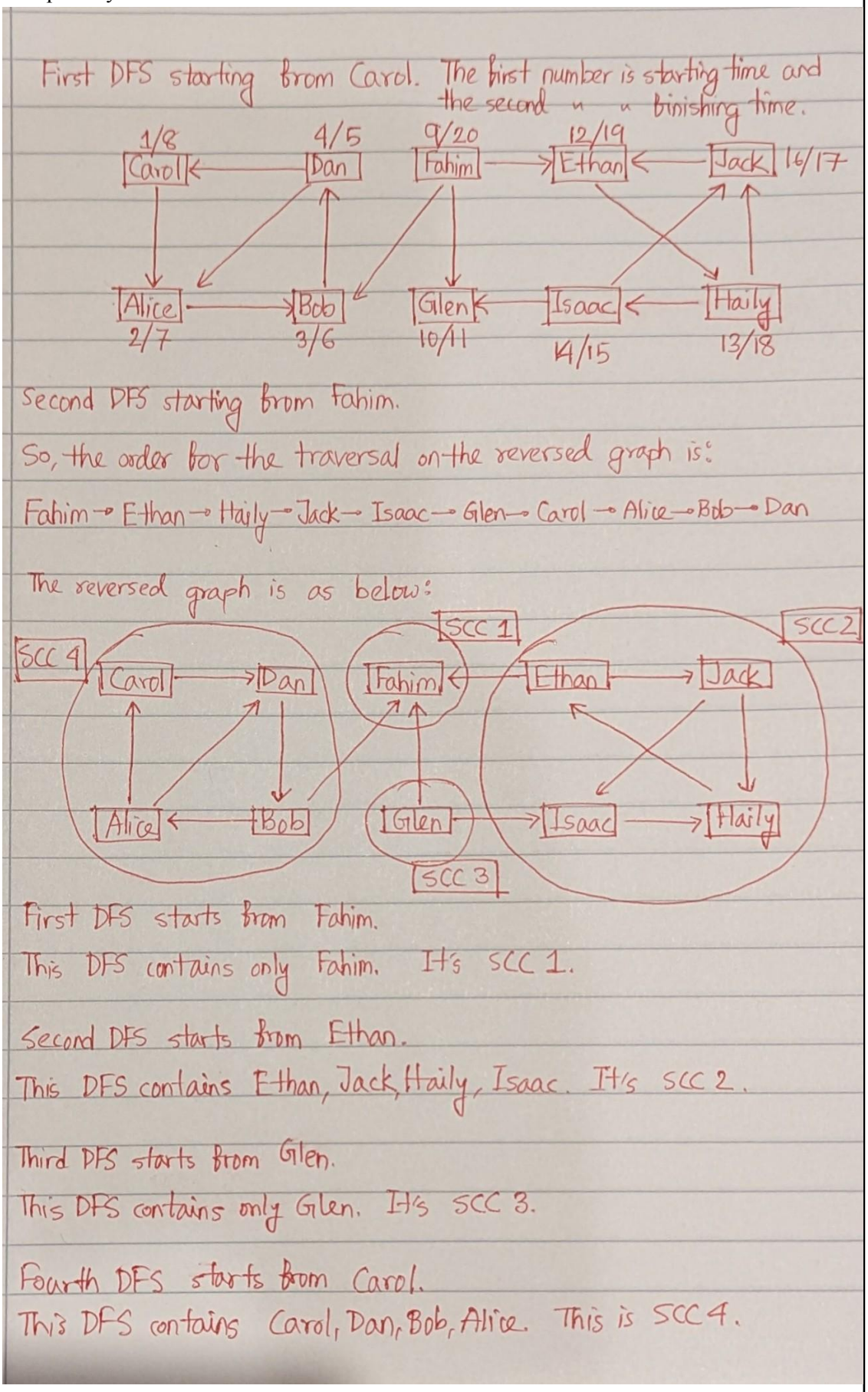
d) [CO3] We also want to find the person who can visit the most houses. **Propose** an algorithm to implement this. Present your algorithm with a pseudocode/flowchart/step-by-step instructions. **03**

Solutions:

a) **Kosaraju**'s or **Tarjan**'s algorithm to find **Strongly Connected Components** in a directed graph.

b) **Kosaraju**'s algorithm is to first run a DFS on the whole graph and push the nodes in a stack just before returning. Then reverse the graph. Then pop the nodes one by one from the stack and run DFS on the reversed graph. The order of the nodes popped is the same order of the nodes according to the descending finishing times from the first round of DFS on the actual graph. As the DFS order can vary, there can be multiple different DFS orders for this solution.

But the SCCs will always be the same. Here one example solution is provided, but there are multiple ways to reach the final answer.



c) **Fahim** and **Glen** - as they cannot form groups according to the definition provided.

d) We can just run a DFS from each node and calculate how many nodes are reachable from each node. The time complexity for this approach will be $O(V * (V+E)) = O(VE) = O(V^3)$. Running DFS only from the nodes which have zero indegree is an optimization.

Or one can first create the SCCs and think of them as nodes. Then the graph will turn into a DAG. Then do the same, run DFS from each node (SCC) and calculate the answer. But it will still have the same time complexity.

There's a better solution in a paper but that is too advanced and out of scope.

Answer only one from 2, 2-OR

2. Consider the following modified ASCII scheme and the encoded message using it.

Character	a	b	c	d	e	f	g	h	i	j
m_ascii	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Message encoded with m_ascii code:
0000 0100 0100 1000 0100 1000 0100 1000 0001 0101 0001 0001 0110 0000 0100 1001 0010 1001

Now answer the following questions.

a) [CO1] **Decode** the message. **01**

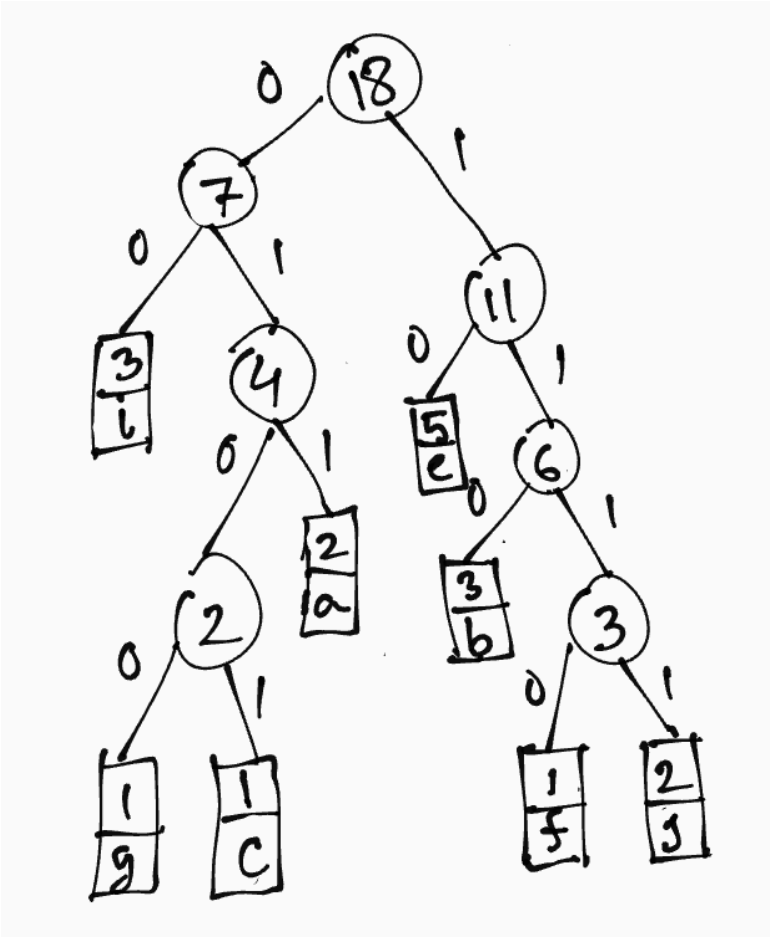
b) [CO1] **Show** simulation to find the encoding of each of the characters using Huffman coding technique. While creating a tree, less frequency nodes go to the left side of the root; left is considered as a 0, right is 1. **06**

c) [CO1] **Calculate** the number of bits required in the encoded message using huffman technique. **03**

Solutions:

a) String after decoding mascii: aeeieieibfbbgaejcj

b) Frequency: {'e': 5, 'i': 3, 'b': 3, 'a': 2, 'j': 2, 'f': 1, 'g': 1, 'c': 1}



c) Character bits = {'e': 2, 'i': 2, 'b': 3, 'a': 3, 'j': 4, 'f': 4, 'g': 4, 'c': 4}

Total bits = 5*2 + 3*2 + 3*3 + 2*3 + 2*4 + 4 + 4 + 4 = 51

OR

2. OR

You Study in a university and there are some faculties in your university and their class schedule (start and end time) are given below. As a pantomath (who wants to learn a lot) person you want to gather more knowledge and do class of the maximum faculties possible. But, due to the time conflict, one student cannot cover all the classes. So, you asked your friends Derke, Leo and Chronicle how you can maximize the number of classes. Derke said you should choose the classes which have the lowest duration and Leo said you should choose those classes which end early and Chronicle said you should pick classes that start early.

a) [CO1] Who's method will you **select** in order to complete the classes of maximum faculties? **01**

b) [CO1] Using the method you mentioned in (a) **simulate** and find out the maximum number of classes you can do from the following schedule and also find out the initials of the faculties. **06**

Faculty	Start Time	End Time
MIBA	1	10
MZU	2	3
AGD	6	8

		MNR	10	11
		RIM	5	7
		FGZ	3	6
		SBD	7	10
	c) [CO1] As one student cannot cover all the classes so you want to determine the minimum number of students needed so that all the classes are covered. **Explain** your algorithm in a pseudocode/ flow-chart/ step-by-step instructions format.		**03**
	Solutions: a) Leo is correct. Early Finish b) Sort the table according to End time Faculty Start Time End Time MZU 2 3 FGZ 3 6 RIM 5 7 AGD 6 8 MIBA 1 10 SBD 7 10 MNR 10 11 So, the faculties will be -> MZU,FGZ,AGD,MNR (Total 4) c) Any step-by-step solution can also be accepted , given the logic is correct. SCHEDULE-INTERVALS(I) $\triangleright$ $I = \{I_i\}, I_i = (s_i, f_i)$ 1 R = Sorted requests in order of starting times, breaking ties arbitrarily, such that $s_i \leq s_j$ when $i < j$. 2 $m \leftarrow 0$ $\triangleright$ the optimal number of resources needed to schedule R 3 **while** $R \neq \emptyset$ 4 **do** *req* = extract the next element in R 5 **if** there is a resource j with no interval conflicting with *req* 6 **then** schedule interval *req* on resource j 7 **else** 8 $m \leftarrow m + 1$ $\triangleright$ allocate a new resource 9 schedule interval *req* on resource m		
3.	Karen started a website that displays different advertisements. She wants to maximize her advertisement revenue. Each month she receives a list of advertisements, each with a certain value		

(revenue) and a size (space taken on the webpage). The webpage has a limited amount of space, and she selects a subset of advertisements to display in order to maximize her total revenue, while staying within the available space. No advertisement is selected more than once, and each advertisement that is selected has to be displayed in full.

For example, following is a list of advertisements she received for the month of August:

Advertisements:	Solution:
Ad 1 - Value: $10, Size: 2 Ad 2 - Value: $8, Size: 1 Ad 3 - Value: $15, Size: 3 Ad 4 - Value: $6, Size: 2 Maximum Space Available: 5	Select Advertisements: Ad 1, Ad 3 Total Revenue: $10 + $15 = $25 Space Occupied = 2 +3 = 5

a) [CO1] Suppose, you are trying to come up with a Dynamic Programming approach to solve this problem. You build a table, each cell representing the solution (optimal revenue value) to a subproblem. **Give** the formula for filling up each entry of the table. **02**

b) [CO1] Suppose in the month of September, she received the following list of advertisements: **06**

Advertisements:
Ad 1 - Value: $12, Size: 4
Ad 2 - Value: $8, Size: 2
Ad 3 - Value: $9, Size: 3
Ad 4 - Value: $7, Size: 2
Ad 5 - Value: $5, Size: 1

The maximum available space is 5.

Fill the following table according to the formula you developed in question (a). Then write the set of advertisements Karen should select.

Value	Size	Item	Maximum Available Space					
			0	1	2	3	4	5
-	-	0						
12	4	1						
8	2	2	0	0	8	8	12	12
9	3	3						
7	2	4						
5	1	5						

c) [CO3] Suppose, Karen discovered that the top half of some of the advertisements contain nothing. So, from the month of October she is implementing a new feature, the size of at most K items can be changed to half of its original size. Revenue will stay the same. For example, **02**

There are four advertisements to select from,
Value = [10, 12, 8, 6]
Required Space = [4, 5, 6, 6]
Max Available Space = 7
K = 1

Output: 22

Explanation: Without changing the required space of any item, the solution would be taking only Ad-2, giving a revenue of 12. But if we change the size of Ad-1 into half of its original size, we can take both Ad-1 and 2. Then the total revenue is 22, which is the maximum.

Think about how you would solve this problem using a Dynamic Programming approach. Then, **write** a recursive formulation of your solution or **explain** your idea in pseudocode/flow-chart/step-by-step instructions format.

Solutions:

(a) $\vartheta(n, W) = \vartheta(n - 1, W)$ **if** $w_n > W$;

$\max(v_n + \vartheta(n - 1, W - w_n), \vartheta(n - 1, W))$

(b)

Value	Size	Item	Maximum Available Space					
			0	1	2	3	4	5
0	0	0	0	0	0	0	0	0
12	4	1	0	0	0	0	12	12
8	2	2	0	0	8	8	12	12
9	3	3	0	0	8	9	12	17
7	2	4	0	0	8	9	15	17
5	1	5	0	5	8	13	15	20

Take items 2,4,5

(c) $\vartheta(n, W, k) = \max(v_n + \vartheta(n - 1, W - w_n, k),$

$v_n + \vartheta(n - 1, W - w_n/2, k-1)$

$\vartheta(n - 1, W, k)$

$)$

Explanation:

In Dynamic programming, we can consider a 3D DP table where the state DP[i][j][k] will denote the maximum value we can obtain if we are considering values from 1 to i-th, weight of the knapsack is j and we can half the weight of at most k values. Basically, we are adding one extra state, the number of weights that can be halved in a traditional 2-D 01 knapsack DP matrix.

Now, three possibilities can take place:

- Include item with full weight if the item's weight does not exceed the remaining weight
- Include the item with half weight if the item's half weight does not exceed the remaining weight
- Skip the item

Now we have to take the maximum of these three possibilities.

- If we do not take the i-th weight then dp[i][j][k] would remain equal to dp[i – 1][j][k], just like traditional knapsack.
- If we include item with half weight then dp[i][j][k] would be equal to dp[i – 1][j – wt[i] / 2][k – 1] + val[i] as after including i-th value our remaining knapsack capacity would be j – wt[i] / 2 and our number of half operations (k) would increase by 1.
- Similarly, if we include item with full weight then dp[i][j][k] would be equal to dp[i – 1][j – wt[i]][k] + val[i] as knapsack capacity in this case would reduce to j – wt[i].

We simply take the maximum of all three choices.

Please Turn Over

4. The white council once decided to install bidirectional rail tracks to connect several realms of middle earth. Upon their request, the elves and the dwarves devised a plan. The following table lists the tracks that can be constructed.

From	To	Length of the track	Construction cost	Construction time
Arnor	Fangorn	25	70	80
Arnor	Lindon	20	40	30
Arnor	Mirkwood	30	80	70
Fangorn	Gondor	10	30	40
Fangorn	Lindon	40	90	110
Fangorn	Mirkwood	20	60	50
Gondor	Harad	25	100	90
Gondor	Lindon	45	110	100
Gondor	Mirkwood	30	50	60
Gondor	Rhun	40	120	130
Harad	Mirkwood	45	150	180
Mirkwood	Rhun	25	180	150

As there is only one engine available for the task, simultaneous construction of multiple tracks is not possible. Going through the plan, the council decided that the following tracks **must be constructed**.

- Gondor to Lindon
- Arnor to Mirkwood

However, due to a financial crisis, the council wants to connect all the realms with **minimum total construction cost**. Now answer the following questions.

a) [CO1] Which other tracks should they select for construction? **Show** a simulation of your procedure. **05**

b) [CO1] **Write** the total construction cost of the selected tracks. **01**

c) [CO1] **Calculate** the total construction time required if no two tracks are constructed simultaneously. **01**

d) [CO1] After these tracks have been constructed, what would be the minimum distance of Rhun from all other realms? Just **write** the distance from each realm. **03**

Solutions:

(a) Simulation

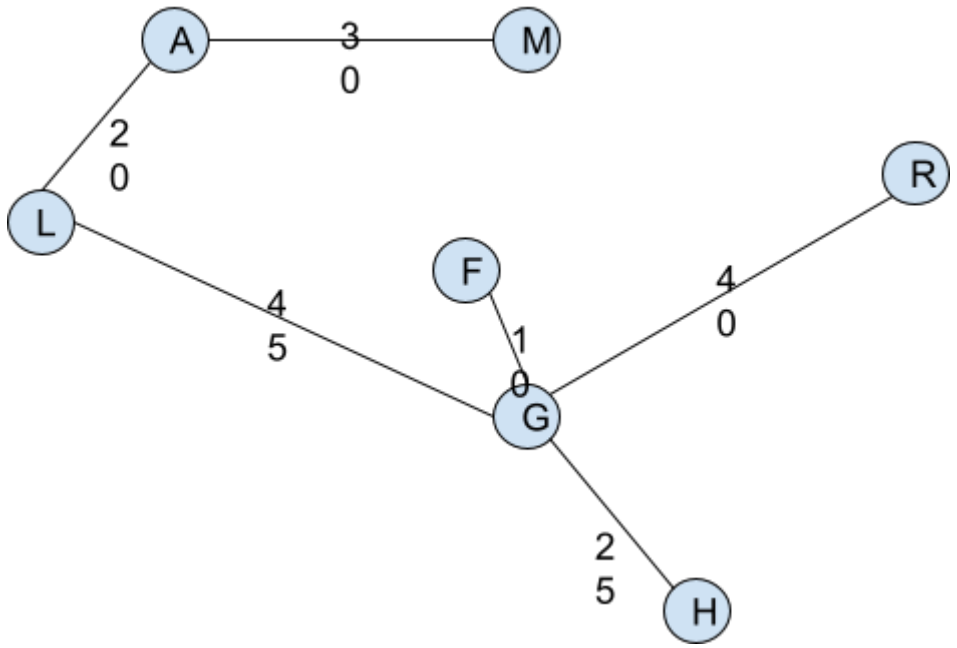
next edge	verdict	vertex sets
		{A}, {F}, {G}, {H}, {L}, {M}, {R}
(G,L) – 110	Must take	{A}, {F}, {G, L}, {H}, {M}, {R}
(A,M) – 80	Must take	{A, M}, {F}, {G, L}, {H}, {R}
(F,G) – 30	Taken	{A, M}, {F, G, L}, {H}, {R}
(A,L) – 40	Taken	{A, F, G, L, M}, {H}, {R}
(G,M) – 50	Cannot take	–
(F,M) – 60	Cannot take	–
(A,F) – 70	Cannot take	–
(A,M) – 80	Already taken	–
(F,L) – 90	Cannot take	–
(G,H) – 100	Taken	{A, F, G, H, L, M}, {R}
(G,L) – 110	Already taken	–
(G,R) – 120	Taken	{A, F, G, H, L, M, R}
(H,M) – 150	–	–
(M,R) – 180	–	–

So, (F,G), (A,L), (G,H) & (G,R) should be selected.

(b) 480

(c) 460

(d) The spanning tree (with track lengths)



Realm	A	F	G	H	L	M
Dist to Rhun	105	50	40	65	85	135

B

BRAC UNIVERSITY
Department of Computer Science and Engineering

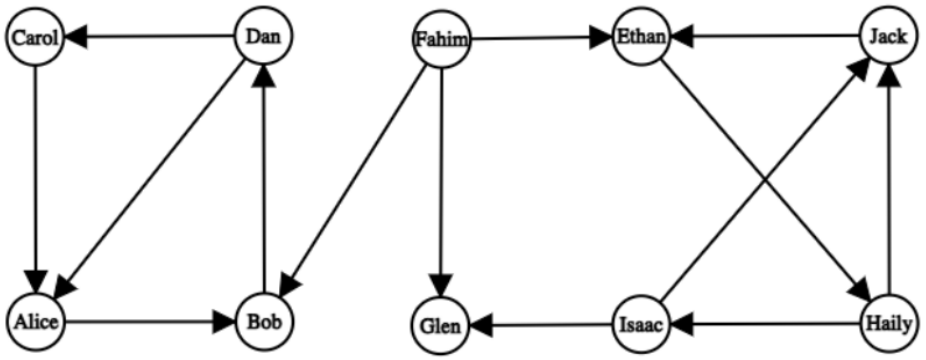
Examination: Final Exam | Semester: Summer 2023
Duration: 1 Hour 40 Minutes | Full Marks: 40

CSE 221: Algorithms

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

Name:	ID:	Section:

1. Among a group of 10 friends, it's not feasible for each individual to visit the houses of all the others. With an upcoming tournament in mind, the friends need to establish groups. The criterion for forming these groups is that a friend must be capable of visiting the houses of every other member in their group. The task involves calculating the count of viable groups following this criterion. A group has to consist of a minimum of two people.



a) [CO1] The end result will showcase the list of groups and the members. **Name** the algorithm tailored for this situation. **01**

b) [CO1] **Show** a simulation of your chosen algorithm using the graph above. **Write** the number of groups that can be formed and the members of each group. **05**

c) [CO1] Those who are incapable of forming a group will be identified as individuals ineligible for tournament participation. **Identify** those people if there are any. **01**

d) [CO3] We also want to find the person who can visit the most houses. **Propose** an algorithm to implement this. Present your algorithm with a pseudocode/flowchart/step-by-step instructions. **03**

Solutions:

Same as the one in Set A.

Answer only one from 2, 2-OR

2. Consider the following modified ASCII scheme and the encoded message using it.

Character	a	b	c	d	e	f	g	h	i	j
m_ascii	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Message encoded with m_ascii code:
0010 0000 0010 0000 0011 0110 0111 1000 0011 1001 0000 0010 0011 0010 1001 1000 0010 0111

Now answer the following questions.

a) [CO1] **Decode** the message. **01**

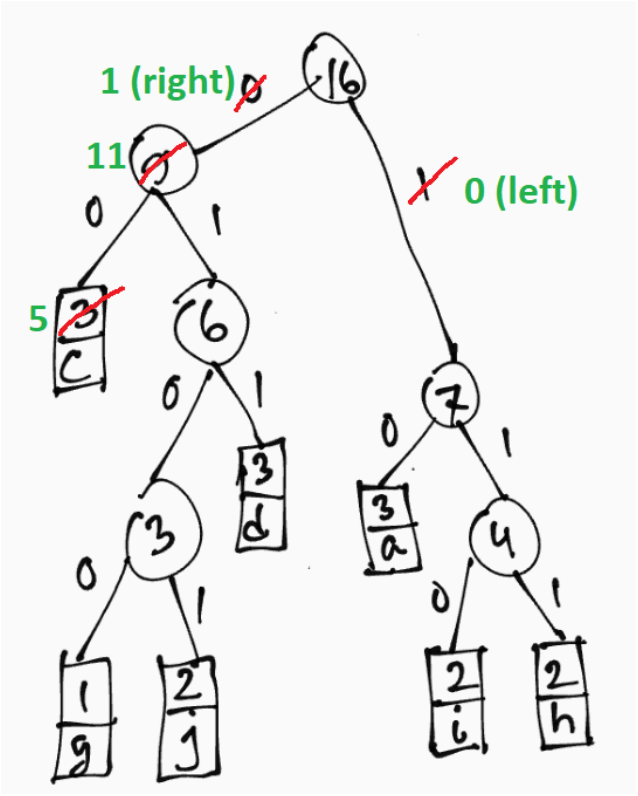
b) [CO1] **Show** simulation to find the encoding of each of the characters using Huffman coding technique. While creating a tree, less frequency nodes go to the left side of the root; left is considered as a 0, right is 1. **06**

c) [CO1] **Calculate** the number of bits required in the encoded message using huffman technique. **03**

Solutions:

a) String after decoding ascii: cacadghidjacdcjich

b) Frequency: {'c': 5, 'a': 3, 'd': 3, 'h': 2, 'i': 2, 'j': 2, 'g': 1}



c) Character bits = {'c': 2, 'a': 2, 'd': 3, 'h': 3, 'i': 3, 'j': 4, 'g': 4}

Total bits = 5*2 + 3*2 + 3*3 + 2*3 + 2*3 + 2*4 + 1*4 = 49

OR

2. OR

You Study in a university and there are some faculties in your university and their class schedule (start and end time) are given below. As a pantomath (who wants to learn a lot) person you want to gather more knowledge and do class of the maximum faculties possible. But, due to the time conflict, one student cannot cover all the classes. So, you asked your friends Derke, Leo and Chronicle how you can maximize the number of classes. Derke said you should choose the classes which have the lowest duration and Leo said you should choose those classes which end early and Chronicle said you should pick classes that start early.

a) [CO1] Who's method will you **select** in order to complete the classes of maximum faculties? **01**

b) [CO1] Using the method you mentioned in (a) **simulate** and find out the maximum number of classes you can do from the following schedule and also find out the initials of the faculties. **06**

Faculty	Start Time	End Time
MIBA	2	11
MZU	3	4
MNR	7	9
AGD	11	12
FGZ	6	8
MNR	4	7
SBD	8	11

c) [CO1] As one student cannot cover all the classes so you want to determine the minimum number of students needed so that all the classes are covered. **Explain** your algorithm in a pseudocode/ flow-chart/ step-by-step instructions format. **03**

Solution :

a) Leo is correct. Early Finish

b) Sort the table according to End time

Faculty	Start Time	End Time
MZU	3	4
MNR	4	7
FGZ	6	8
MNR	7	9
MIBA	2	11
SBD	8	11
AGD	11	12

So, the faculties will be -> MZU,MNR,MNR,AGD (Total 4)

c) Any step-by-step solution can also be accepted , given the logic is correct.

```
SCHEDULE-INTERVALS(I)      ▷ I = {I_i}, I_i = (s_i, f_i)
1  R = Sorted requests in order of starting times, breaking ties
   arbitrarily, such that s_i ≤ s_j when i < j.
2  m ← 0 ▷ the optimal number of resources needed to schedule I
3  while R ≠ ∅
4        do req = extract the next element in R
5           if there is a resource j with no interval conflicting with
6              then schedule interval req on resource j
7              else
8                   m ← m + 1          ▷ allocate a new resource
9                   schedule interval req on resource m
```

3. Karen started a website that displays different advertisements. She wants to maximize her advertisement revenue. Each month she receives a list of advertisements, each with a certain value (revenue) and a size (space taken on the webpage). The webpage has a limited amount of space, and she selects a subset of advertisements to display in order to maximize her total revenue, while staying within the available space. No advertisement is selected more than once, and each advertisement that is selected has to be displayed in full.

For example, following is a list of advertisements she received for the month of August:

Advertisements:	Solution:
Ad 1 - Value: \$10, Size: 2 Ad 2 - Value: \$8, Size: 1 Ad 3 - Value: \$15, Size: 3 Ad 4 - Value: \$6, Size: 2 Maximum Space Available: 5	Select Advertisements: Ad 1, Ad 3 Total Revenue: \$10 + \$15 = \$25 Space Occupied = 2 +3 = 5

a) [CO1] Suppose, you are trying to come up with a Dynamic Programming approach to solve this problem. You build a table, each cell representing the solution (optimal revenue value) to a subproblem. **Give** the formula for filling up each entry of the table. **02**

b) [CO1] Suppose in the month of September, she received the following list of advertisements: **06**

Advertisements:
Ad 1 - Value: \$1, Size: 1
Ad 2 - Value: \$4, Size: 3
Ad 3 - Value: \$5, Size: 4
Ad 4 - Value: \$7, Size: 5
The maximum available space is 7.

Fill the following table according to the formula you developed in question (a). Then write the set of advertisements Karen should select.

Value	Size	Item	Maximum Available Space							
			0	1	2	3	4	5	6	7

0	0	0								
1	1	1								
4	3	2	0	1	1	4	5	5	5	5
5	4	3								
7	5	4								

c) [CO3] Suppose, Karen discovered that the top half of some of the advertisements contain nothing. So, from the month of October she is implementing a new feature, the size of at most K items can be changed to half of its original size. Revenue will stay the same. For example, **02**

There are four advertisements to select from,
Value = [17, 20, 10, 15]
Required Space = [4, 2, 7, 5]
Max Available Space = 4
K = 1

Output: 37
Explanation: Without changing the required space of any item, the solution would be taking only Ad-2, giving a revenue of 20. But if we change the size of Ad-1 into half of its original size, we can take both Ad-1 and 2. Then the total revenue is 37, which is the maximum.

Think about how you would solve this problem using a Dynamic Programming approach. Then, **write** a recursive formulation of your solution or **explain** your idea in pseudocode/flow-chart/step-by-step instructions format.

(a) Same as set A

$$\vartheta(n, W) = \begin{cases} \vartheta(n-1, W) & \text{if } w_n > W; \\ \max(v_n + \vartheta(n-1, W - w_n), \vartheta(n-1, W)) \end{cases}$$

(b)

Value	Size	Item	Maximum Available Space							
			0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	1	1	1	1
4	3	2	0	1	1	4	5	5	5	5
5	4	3	0	1	1	4	5	6	6	9
7	5	4	0	1	1	4	5	7	8	9

Take items 2,3

(c) Same as set A

$$\vartheta(n, W, k) = \max\begin{cases} v_n + \vartheta(n-1, W - w_n, k), \\ v_n + \vartheta(n-1, W - w_n/2, k-1) \\ \vartheta(n-1, W, k) \end{cases}$$

4. The white council once decided to install bidirectional rail tracks to connect several realms of middle earth. Upon their request, the elves and the dwarves devised a plan. The following table lists the tracks that can be constructed.

From	To	Length of the track	Construction cost	Construction time
Arnor	Fangorn	25	70	80
Arnor	Lindon	20	40	30
Arnor	Mirkwood	30	80	70
Fangorn	Gondor	10	30	40
Fangorn	Lindon	40	90	110
Fangorn	Mirkwood	20	60	50
Gondor	Harad	25	100	90
Gondor	Lindon	45	110	100
Gondor	Mirkwood	30	50	60
Gondor	Rhun	40	120	130
Harad	Mirkwood	45	150	180
Mirkwood	Rhun	25	180	150

As there is only one engine available for the task, simultaneous construction of multiple tracks is not possible. Going through the plan, the council decided that the following tracks **must be constructed**.

- Mirkwood to Rhun
- Fangorn to Lindon

As the election is nearby, the council wants to connect all the realms with **minimum total construction time**. Now answer the following questions.

a) [CO1] Which other tracks should they select for construction? **Show** a simulation of your procedure. **05**

b) [CO1] **Write** the total construction time of the selected tracks. No two tracks are constructed simultaneously. **01**

c) [CO1] **Calculate** the total construction cost required. **01**

d) [CO1] After these tracks have been constructed, what would be the minimum distance of Harad from all other realms? Just **write** the distance from each realm. **03**

Solutions:

(a) Simulation

next edge	verdict	vertex sets

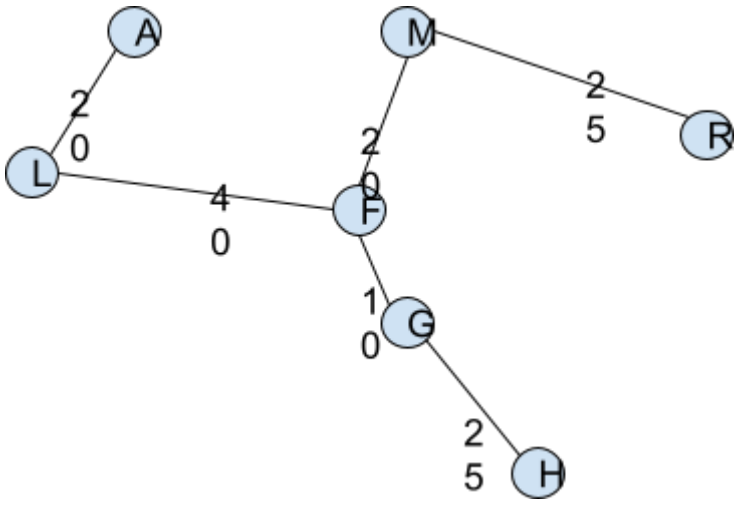
		{A}, {F}, {G}, {H}, {L}, {M}, {R}
(M,R) – 150	Must take	{A}, {F}, {G}, {H}, {L}, {M, R}
(F,L) – 110	Must take	{A}, {F, L}, {G}, {H}, {M, R}
(A,L) – 30	Taken	{A, F, L}, {G}, {H}, {M, R}
(F,G) – 40	Taken	{A, F, G, L}, {H}, {M, R}
(F,M) – 50	Taken	{A, F, G, L, M, R}, {H}
(G,M) – 60	Cannot take	–
(A,M) – 70	Cannot take	–
(A,F) – 80	Cannot take	–
(G,H) – 90	Taken	{A, F, G, H, L, M, R}
(G,L) – 100	–	–
(F,L) – 110	Already taken	–
(G,R) – 120	–	–
(M,R) – 150	Already taken	–
(H,M) – 180	–	–

So, (A,L), (F,G), (F,M) & (G,H) should be selected.

(b) 470

(c) 500

(d) The spanning tree (with track lengths)



Realm	A	F	G	L	M	R
Dist to Harad	95	35	25	75	55	80