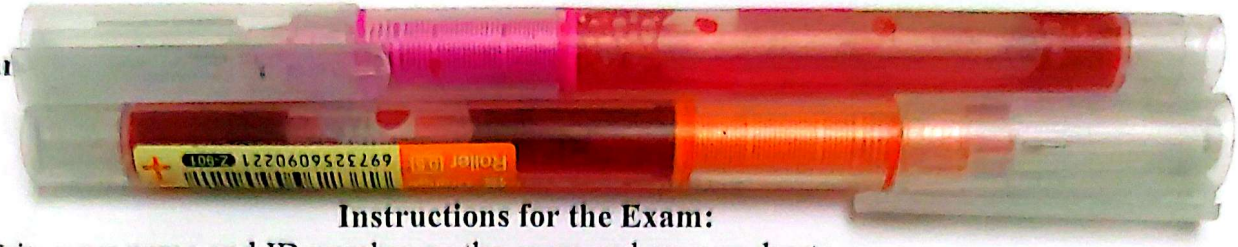
University of Petra		جامعة البترا - ثلاثون عاما University of Petra
Faculty of Information Technology		كلية تكنولوجيا المعلومات
Department of Computer Science	جامعة البترا	قسم علوم الحاسوب

Advanced Algorithms
601326
Final Exam – 2025 1

Your Nar



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	5
Q2			5	5
Q3			2	0.5
Q4	I2		4	4
Q5			5	5
Q6			5	5
Q7	I2		4	3.5
Q8			3	1.5
Q9			7	5
Total			40	34.5

This exam has 9 Questions. The total mark is 40

5

Question 1) Choose the correct answer for each of the following: **(5 marks)**

1. What is time complexity of TSP using Branch and Bound in the worst case? (where n is number of vertices and E is number of edges)
   a) O(1)
   b) O(n*E)
   c) $O(n^2)$
   d) O(n!)
2. What is the space complexity of the Insertion Sort algorithm?
   a) O(1)
   b) $O(n^2)$
   c) O(n)
   d) O(n log n)
3. Given Build_Max_Heap Algorithm:

```
Build_Max_Heap(A){
   n= length(A)
   for i= n/2 down to 1
      max_heapify(A, i, n)
}
```

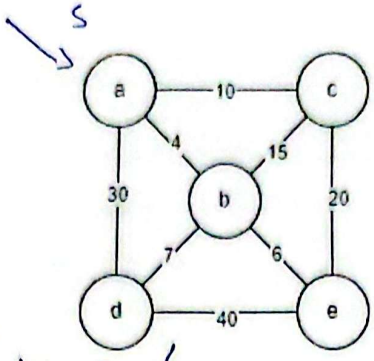
Based on your understanding of the algorithm, why does i start from n/2 ?

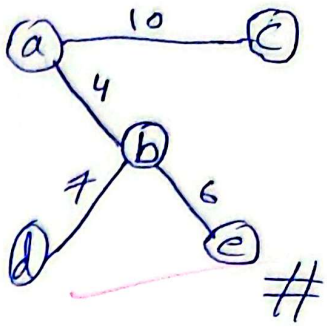
   a) to get value of non-leaf vertices
   b) to get index of non-leaf vertices
   c) to divide A into two sub-arrays and sort each one separately
   d) to call max_heapify only twice
   e) none of the above

4. Which of the following algorithms design techniques support backtracking:
   a) Divide and conquer
   b) Greedy programming
   c) Dynamic programming
   d) None of the above

5. What is the time complexity of the Dynamic algorithm of 0/1 Knapsack Problem? (where n is number of items, and W is knapsack capacity)
   a) O(n+W)
   b) O(n*W)
   c) O(n log W)
   d) $O(n^2)$

Question 2) Apply **Prim's** algorithm to the following graph to find **MST** showing detailed steps. **(5 marks)** 5





#

h = { ~~4~~, 10, 30 }

= { 6, 7, 10, 15, 30 }

= { ~~7~~, ~~10~~, ~~15~~, ~~20~~, ~~30~~, ~~40~~ }

cost of MST = 27

Question 3) Based on your understanding of Floyd's algorithm (for weighted graphs), write a recursive relation for finding R^k_{ij}. **(2 marks)** 0.5

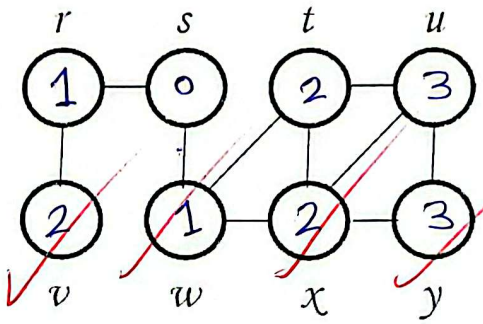
$$R^k_{ij} = \begin{cases} R_{ij} & , k = 0 \\ R_{ik} + R_{kj} & , R_{ik} + R_{kj} > R_{ij} \\ R_{ij} & , R_{ik} + R_{kj} < R_{ij} \end{cases}$$

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

	Breadth-First Search	Quick Sort	Selection Sort	Recursive Towers of Hanoi
Best Case	$\Omega(V+E)$	$\Omega(n \log n)$	$\Omega(n^2)$	2^n
Worst Case	$O(V+E)$	$O(n^2)$	$\Theta(n^2)$	2^n

4

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)**



5

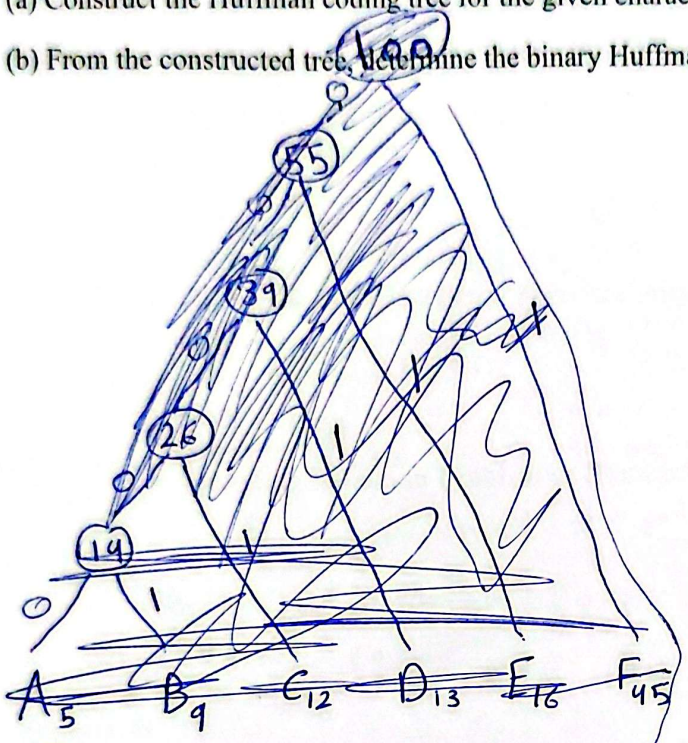
Question 6) Given the following set of characters and their frequencies: **(5 marks)**

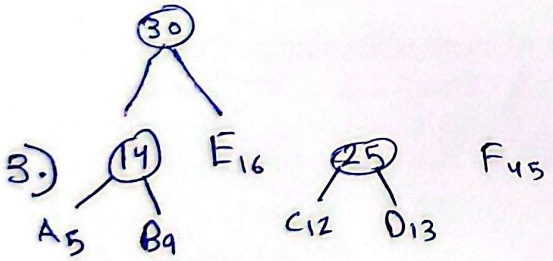
Character	A	B	C	D	E	F
Frequency	5	9	12	13	16	45

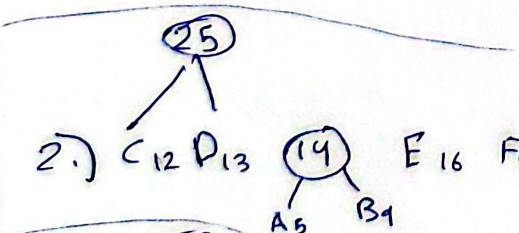
(a) Construct the Huffman coding tree for the given characters by clearly showing details

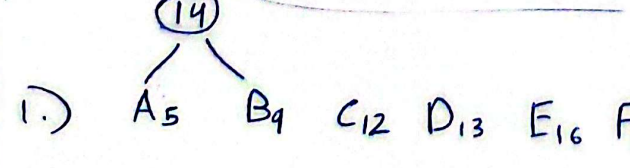
(b) From the constructed tree, determine the binary Huffman code for each character.

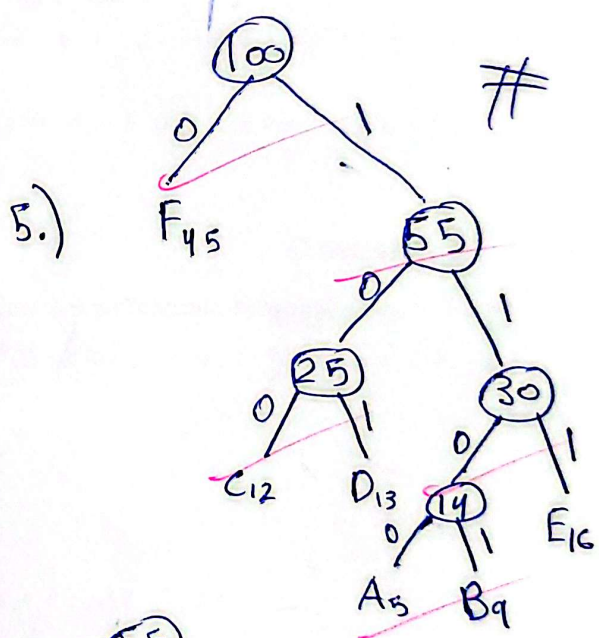
letter	code
A	1100
B	1101
C	100
D	101
E	111
F	0

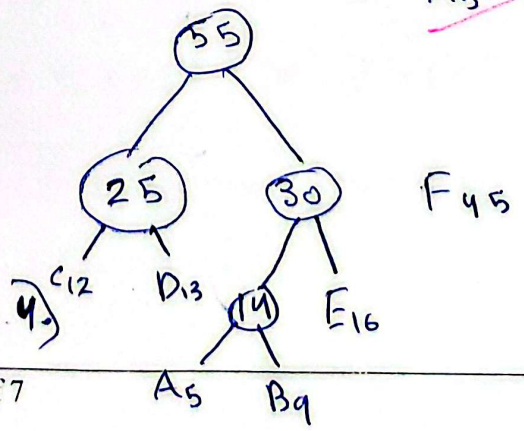












3.5

Question 7) Discuss **briefly (two sentences maximum)** the **difference** between the following: **(4 marks)**

- **P vs NP problems:**

P is a class for problems that is ~~easily~~ solved fast

1 NP is a class for problems hard to solve but easy to verify an answer.

- **Decision vs Optimization problems:**

1 Decision: is simple problems with a simple "Yes" or "No" answer, but Optimization is trying to find the best optimal solution

- **Branch & Bound vs Brute Force techniques:**

0.5 Branch & Bound. divide the problem into branches and check every posibility to find the best solution but its hard to solve, Brute force checks every posibility to find the answer its easy to solve but not ~~eff~~ efficient.

- **Traveling Salesman Problem vs Hamiltonian Cycle:**

1 TSP gives shortest path to complete the cycle.
but Hamiltonian doesn't care for the length of the path.

Question 8) **(3 marks)**

Write the recursive formulation for the 0/1 Knapsack Problem for a Dynamic Programming solution. n = # items / W = Capacity / w_i = weight of item i

V_i = benefit of i

1.5

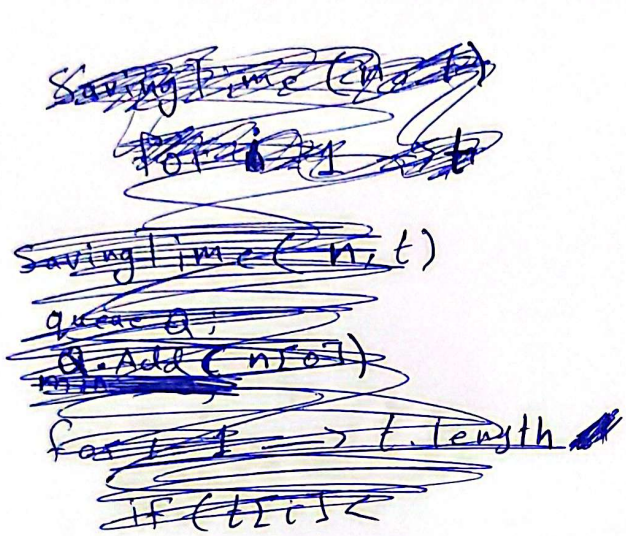
$$V[i,w] = \begin{cases} 0 & , i=0 \text{ or } w=0 \\ V_i + V[i-1, w] & , V_i + V[i-1, w] > V[i-1, w-w_i] \\ V[i-1, w-w_i] & , V_i + V[i-1, w] < V[i-1, w-w_i] \end{cases}$$

Question 9) **(7 marks)**

Consider the problem of scheduling n jobs of known durations t1, ..., tn for execution by a single processor. The jobs can be executed in any order, one job at a time. You want to find a schedule that minimizes the total time spent by all the jobs in the system. (The time spent by one job in the system is the sum of the time spent by this job in waiting plus the time spent on its execution.)

a) Design a greedy algorithm for this problem.
b) Does your greedy algorithm always yield an optimal solution?

a) sol



let t = array of durations needed for each job.

priority queue

Savetime(t)
priority queue Q;
for i = 1 ⟶ t
Q.Add(i)

for i = 1 ⟶ Q.length # execution
Q.remove(i)

Good Luck

b) Yes, because when I start with the minimal time the waiting time for other jobs will be less.