Question 1 [20 POINTS]

Ahmed and Wasay, two of our TAs, are planning their schedules for a busy morning with five tasks to complete.

Task	Time	Description
F (Food)	1 Hour	Pick up food for the groups research seminar,
H (Homework)	2 consecutive hours	Prepare homework questions
P (PR2 Robot)	1 Hour	Prepare PR2 robot for a group of preschoolers visit,
S (seminar)	1 Hour	Lead the research seminar
T (Teach)	2 consecutive hours	Teach the preschoolers about the PR2 robot

The schedule consists of one-hour slots: **8am-9am, 9am-10am, 10am-11am, 11am-12pm**. The requirements for the schedule are as follows:

a	In any given time slot each TA can do at most one task (F, H, P, S, T).
b	The PR2 preparation (P) should happen before teaching the preschoolers (T).
c	The food should be picked up (F) before the seminar (S).
d	The seminar (S) should be finished by 10am.
e	Ahmed is going to deal with food pick up (F) since he has a car.
f	The TA not leading the seminar (S) should still attend, and hence cannot perform another task (F, T, P, H) during the seminar.
g	The seminar (S) leader does not teach the preschoolers (T).
h	The TA who teaches the preschoolers (T) must also prepare the PR2 robot (P).

i	Preparing homework questions (H) takes 2 consecutive hours, and hence should start at or before 10am.
j	Teaching the preschoolers (T) takes 2 consecutive hours, and hence should start at or before 10am.

- To formalize this problem as a CSP, use the variables F, H, P, S and T.
- The values they take on indicate the TA responsible for it, and the starting time slot during which the task is carried out (for a task that spans 2 hours, the variable represents the starting time, but keep in mind that the TA will be occupied for the next hour also- make sure you enforce constraint (a)!).
- Hence there are eight possible values for each variable, which we will denote by A8, A9, A10, A11, W8, W9, W10, W11, where the letter corresponds to the TA and the number corresponds to the time slot. For example, assigning the value of A8 to a variables means that this task is carried about by Ahmed from 8am to 9am.

a. Which of the statements above include unary constraints? [2]

d,e,i,j

+0.5 for 1 each correct statement, -0.5 for each incorrect statement

b. In the table below, enforce all **unary constraints** by crossing out values in the table on the below.[6]

F	A8	A9	A10	A11	~~W8~~	~~W9~~	~~W10~~	~~W11~~
H	A8	A9	A10	~~A11~~	W8	W9	W10	~~W11~~
P	A8	A9	A10	A11	W8	W9	W10	W11
S	A8	A9	~~A10~~	~~A11~~	W8	W9	~~W10~~	~~W11~~
T	A8	A9	A10	~~A11~~	W8	W9	W10	~~W11~~

+0.5 marks for correct crossing out each cell ,- 0.5 deduction for incorrect crossing out

c. Start from the table above, select the variable S and assign the value A9 to it. Perform **forward checking** by crossing out values in the table below. [6]

F	A8	~~A9~~	~~A10~~	~~A11~~	~~W8~~	~~W9~~	~~W10~~	~~W11~~
H	~~A8~~	~~A9~~	A10	~~A11~~	~~W8~~	~~W9~~	W10	~~W11~~
P	A8	~~A9~~	A10	A11	W8	~~W9~~	W10	W11
S	~~A8~~	A9	~~A10~~	~~A11~~	~~W8~~	~~W9~~	~~W10~~	~~W11~~
T	~~A8~~	~~A9~~	~~A10~~	~~A11~~	~~W8~~	~~W9~~	W10	~~W11~~

1 marks for each correct row , +1 marks for attempting this part

d. We return to the result from enforcing just the unary constraints, which we did in (c). Select the variable S and assign the value A9. Enforce **arc consistency** (ac3) by crossing out values in the table below. [5]

F	A8	~~A9~~	~~A10~~	~~A11~~	~~W8~~	~~W9~~	~~W10~~	~~W11~~
H	~~A8~~	~~A9~~	A10	~~A11~~	~~W8~~	~~W9~~	~~W10~~	~~W11~~
P	~~A8~~	~~A9~~	~~A10~~	~~A11~~	W8	~~W9~~	~~W10~~	~~W11~~
S	~~A8~~	A9	~~A10~~	~~A11~~	~~W8~~	~~W9~~	~~W10~~	~~W11~~
T	~~A8~~	~~A9~~	~~A10~~	~~A11~~	~~W8~~	~~W9~~	W10	~~W11~~

1 marks for each correct row

e. Compare your answers to (d) and to (c). Does arc consistency remove more values than forward checking? [1]

Yes (1 mark for yes, else 0)

Q2 [20]: Solve the Traveling Salesman Problem (TSP) for 6 cities using a Genetic Algorithm (GA). The cities are labeled $C_1, C_2, \ldots, C_6$ and their pairwise distances are given below (symmetric matrix):

	C_1	C_2	C_3	C_4	C_5	C_6
C_1	0	12	8	20	15	22
C_2	12	0	14	25	30	9
C_3	8	14	0	10	17	28
C_4	20	25	10	0	7	12
C_5	15	30	17	7	0	9
C_6	22	9	28	12	9	0

Parents:

- Parent 1: $[C_1,C_3,C_5,C_2,C_4,C_6]$
- Parent 2: $[C_6,C_2,C_4,C_1,C_3,C_5]$

Tasks:

a. **Uniform Crossover [6]:** Apply the following subset pattern

ABBAAB (A = Parent 1, B = Parent 2)

to generate two children. Resolve duplicate cities by replacing them with the ***missing cities <u>in the order they appear</u> in the other parent.***

- For Child 1, see the order of missing cities in Parent 2 and for Child 2, see the order of the missing cities in Parent 1.
- Finally replace the duplicate (second occurrence of same city) with the missing city in order from relevant parent.

b. **Mutation [4]:** Perform a swap mutation on both children by exchanging the cities at positions 3 and 5.

c. **Fitness Calculation [8]:** Compute the total distance (fitness) for all parents and mutated children.

d. **Conclusion [2]:** What is the change in total distance (fitness) between the worst and the best chromosome? Write only the value.

Solution:

1. Uniform Crossover

Pattern: A B B A A B (positions 1–6, where **A = Parent 1**, **B = Parent 2**).

- **Child 1**:
  - Initial: [C_1 (A), C_2 (B), C_4 (B), C_2 (A), C_4 (A), C_5 (B)]
  - Duplicates: C_2 (positions 2 and 4), C_4 (positions 3 and 5).
  - Missing cities (from Parent 2's order): C_6,C_3.
  - Resolution: Replace C_2 (position 4) with C_6, C_4 (position 5) with C_3.
  - Final Child 1: [C_1, C_2, C_4, C_6, C_3, C_5].
- **Child 2**:
  - **Correct Approach**: Use the **same pattern** (ABBAAB) but swap A/B roles (A = Parent 2, B = Parent 1).
  - Initial: [C_6 (A), C_3 (B), C_5 (B), C_1 (A), C_2 (A), C_4 (B)]
  - Duplicates: C_3 (positions 2 and 5), C_6 (positions 1 and 6).
  - Missing cities (from Parent 1's order): C_2,C_4
  - Resolution: Replace C_3 (position 5) with C_2, C_6 (position 6) with C_4.
  - Final Child 2: [C_6, C_3, C_5, C_1, C_2, C_4].

2. Mutation

- **Child 1**: Swap positions 3 and 5 → [C_1, C_2, C_3, C_6, C_4, C_5].
- **Child 2**: Swap positions 3 and 5 → [C_6, C_3, C_2, C_1, C_5, C_4].

3. Fitness Calculation

Parent 1: [C_1,C_3,C_5,C_2,C_4,C_6]

- Distance: 8(C1→C3)+17(C3→C5)+30(C5→C2)+25(C2→C4)+12(C4→C6)+22(C6→C1)=1148($C1$→$C3$)+17($C3$→$C5$)+30($C5$→$C2$)+25($C2$→$C4$)+12($C4$→$C6$)+22($C6$→$C1$)=114.

Parent 2: [C_6,C_2,C_4,C_1,C_3,C_5]

- Distance: 9(C6→C2)+25(C2→C4)+20(C4→C1)+8(C1→C3)+17(C3→C5)+9(C5→C6)=889(*C*6→*C*2)+25(*C*2→*C*4)+20(*C*4→*C*1)+8(*C*1→*C*3)+17(*C*3→*C*5)+9(*C*5→*C*6)=88.

Mutated Child 1: [C_1, C_2, C_3, C_6, C_4, C_5]

Distance: 12(C1→C2)+14(C2→C3)+28(C3→C6)+12(C6→C4)+7(C4→C5)+15(C5→C1)=8812(*C*1→*C*2)+14(*C*2→*C*3)+28(*C*3→*C*6)+12(*C*6→*C*4)+7(*C*4→*C*5)+15(*C*5→*C*1)=88.

Mutated Child 2: [C_6, C_3, C_2, C_1, C_5, C_4].

- Distance: 28(C6→C3)+14(C3→C2)+12(C2→C1)+15(C1→C5)+7(C5→C4)+12(C4→C6)=8828(*C*6→*C*3)+14(*C*3→*C*2)+12(*C*2→*C*1)+15(*C*1→*C*5)+7(*C*5→*C*4)+12(*C*4→*C*6)=88.

Final Results

Individual	**Fitness (Total Distance)**
Parent 1	114
Parent 2	88
Mutated Child 1	88
Mutated Child 2	88

4. 114 – 88 = 26

Q3: Adversarial Search Problem: Chess game Strategy **[10 Marks]**

You are given a simplified chess endgame position with Black to move. The board consists of only kings and pawns as shown below (W = White pieces, B = Black pieces, W* = White King, B* = Black King):

	1	2	3	4	5
5					B*
4			W*		
3	B				
2		W			
1			W		
Cost		-12			

Where:

- W represents White's pieces (a king at position (3,4) and pawns at positions (2,2) and (3,1))
- B represents Black's pieces (a king at position (5,5) and a pawn at position (1,3))
- Each player can move their king one square up, down, left, right **(diagonal not allowed)**.
- Pawns can only move **one square forward** at a time (Toward's other player).
- A player wins by capturing the opponent's king

Evaluation Function

The following static evaluation function is provided (higher values favor White means White is max player and black is min):

- -10 for each White pawn on the board
- +10 for each Black pawn on the board
- -1 for each square that White's king can move to
- +1 for each square that Black's king can move to
- -5 if White's king threatens Black's king (is one move away)
- +5 if Black's king threatens White's king

Example: Start Position Evaluation

First, let's evaluate the starting position:

- White has 2 pawns: -20
- Black has 1 pawn: +10
- White's king has 4 legal moves: -4
- Black's king has 2 legal moves: +2
- No kings are threatening each other: +0

- Total evaluation: -12

You have to consider the game tree to depth 3 (Black's move, White's response, Black's second move). And answer the below questions accordingly.

For example, let's analyze Black's possible three moves, from the start state:

- Black's king to (4,4) and (5,4).
- Black's pawn to (1,2).

a. Complete the below board after each of these above three moves of black. Also calculate the cost using the evaluation function according to new board position. [6 Marks]

If B* moves to (4,5)					
	1	2	3	4	5
5				B*	
4			W*		
3	B				
2		W			
1			W		
Cost	-11				

If B* moves to (5,4)					
	1	2	3	4	5
5					
4			W*		B*
3	B				
2		W			
1			W		
Cost	-11				

If B moves to (1,4)					
	1	2	3	4	5
5					B*
4	B		W*		
3					
2		W			
1			W		
Cost	-12				

b. For each of these three moves, how many could be the total next possible moves of white? [1 Mark]

6 + 6 + 6 = 18

And next for each move of white how many will be the possible moves of black? [1 Mark]

3+3+4+4+4+4=22, 3+4+4+4+4+4=23, 3+3+3+3+3+3=18; Total=63

c. Mention the time and space complexity for the above tree (means only these three steps Black's move, White's move, black's move). [2 Marks]

Time Complexity	O(b^d) = worst O(12,167), best O(84)
Space Complexity	O(d) = O(3)

Hint: You have no need to draw complete tree and calculate the costs of any other state to answer this. You can think it in your mind or do rough work. One other hint, if you didn't understand this question go-ahead and solve next minimax question first.

Q4: Minimax and Alpha-Beta Pruning **[10 Marks]**

Perform the MiniMax and Alpha Beta Pruning on the given tree. Mention the cost inside empty circles and simply put cross on the pruned edge also bold it.

Note: Count each node when you are visiting it first time, do not count while you are revisiting it.

Algorithm	**Root Node Value**	**# of Nodes Visited**	**# of nodes Pruned**
MiniMax	11	31 (32 if root counts)	0
Alpha-Beta	11	23 (24 if root counts)	8