HW2 - Part A & B

Q1 a)

Hill Climbing for a TSP works as follows:

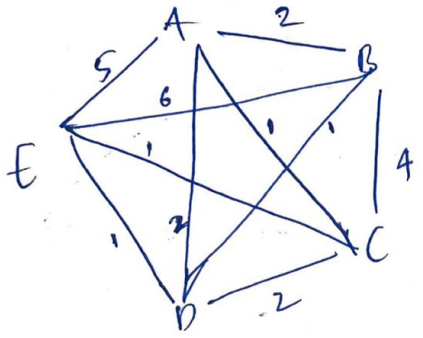
1. As all the cities are connected, generate a sequence of non-repeating cities randomly for the initial state
2. Generate successive states by interchanging two adjacent cities in the sequence
3. Select the state from the previous step with the least cost among all the other states
4. Repeat steps 2 and 3 until the cost cannot be lowered.

Consider the following table 1:

	A	B	C	D	E
A	-	2	1	2	5
B	2	-	4	1	6
C	1	4	-	2	1
D	2	1	2	-	1
E	5	6	1	1	-

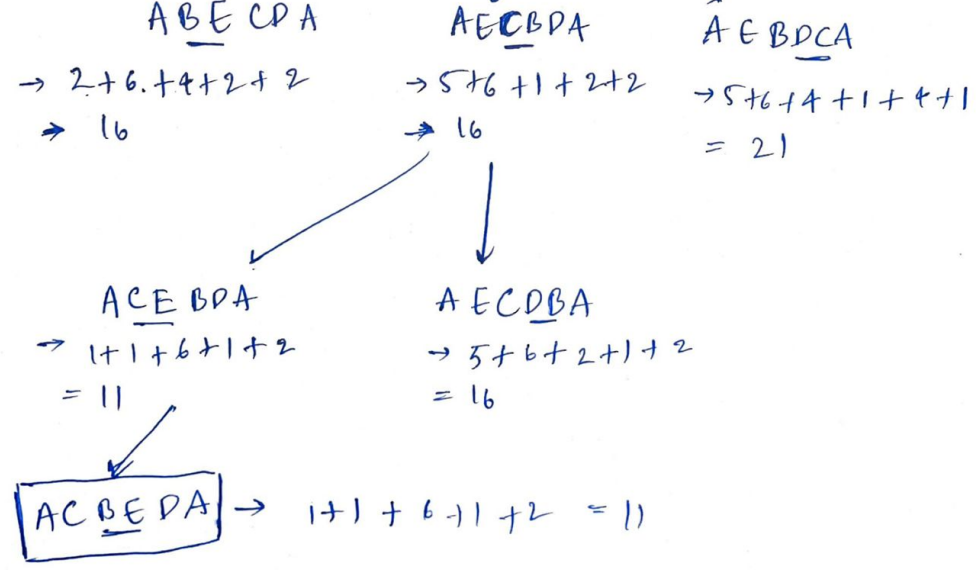
- Each cell corresponds to the cost of the travel between two cities

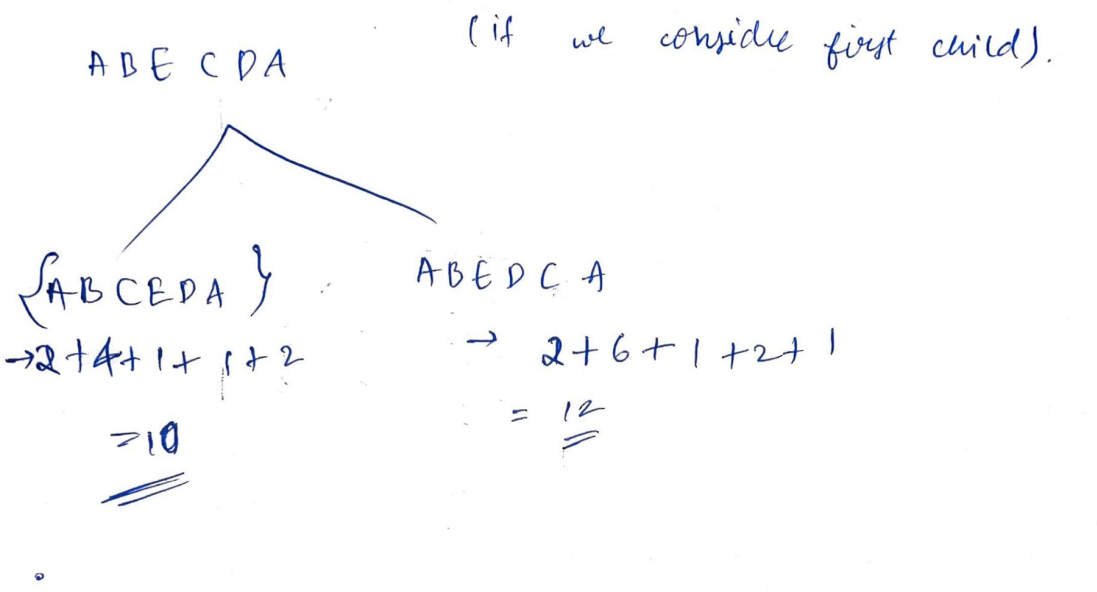
The fully connected graph looks like:



Let's consider a random sequence, A E B C D A where A is the start and the end point.

AEBCD A → (5 + 6 + 4 + 2 + 2) =





- We can see that the Hill climbing Algorithm returns the sequence, ABCEDA with the cost of 10 but ABCEDA is not the actual path of minimum cost path.
- This demonstrates the weakness of the hill climbing approach. The algorithm depends on the initial state as converges to a local minima.

Q1 b)

Simulated Annealing for a TSP works as follows:

1. Generate a sequence of non-repeating cities randomly for the initial state
2. Generate a successive state by interchanging two adjacent cities in the sequence
3. If the state obtained in the step 2 has a lower cost than the previous state, assign it to current state. Else, assign it to the current state with a probability p which is less than one, where p is given by the formula, $p = e^{(\frac{\Delta E}{T})}$.
4. Update p by decreasing the temperature slightly. $T = T - \Delta T$
5. Repeat 2,3 and 4 until convergence.

Note:

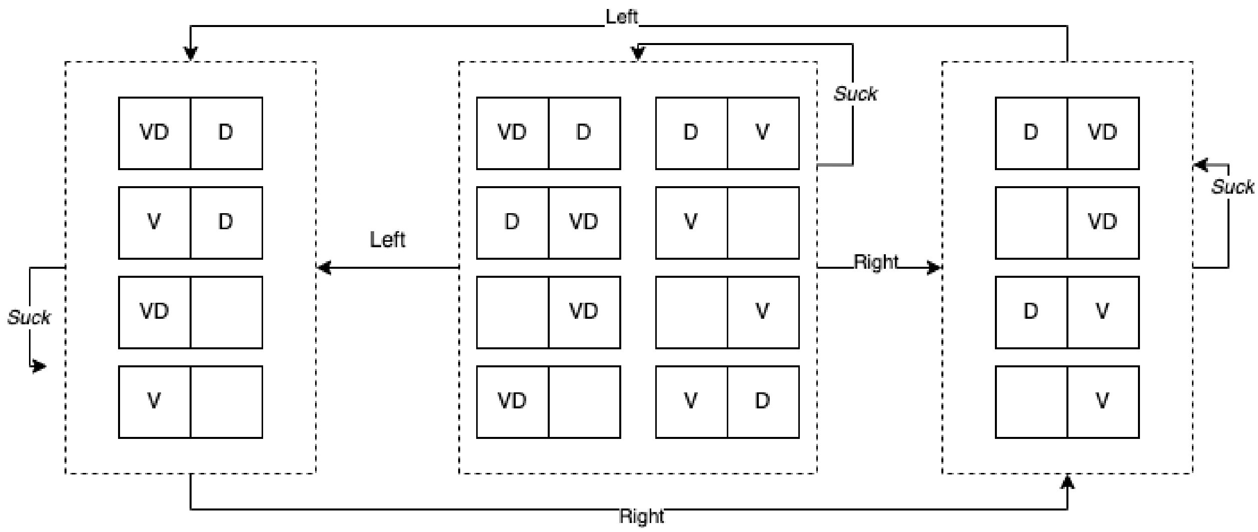
- The probability decreases exponentially with the badness of the sequence
- The amount ΔE by which the evaluation is worsened i.e. (Cost).
- The probability also decreases as the temperature T goes down.

Q2.

This problem poses as a large-scale optimization task with a lot of constraints. For the problem to be solved by simulated annealing, it can be formulated as follows:

- Space is defined by
  - Assembly pieces
  - Configuration pieces (joint angles)
- The linkage and joint angles exactly determine the physical layout of the track. Changing one angle may force changes in others, and the changes will vary depending on whether the other pieces are at their joint-angle limit.
- Evaluation metrics
  - Penalty for tracks lying on each other
  - Penalty for open ends(if)
  - Score for contiguous path

Q3.



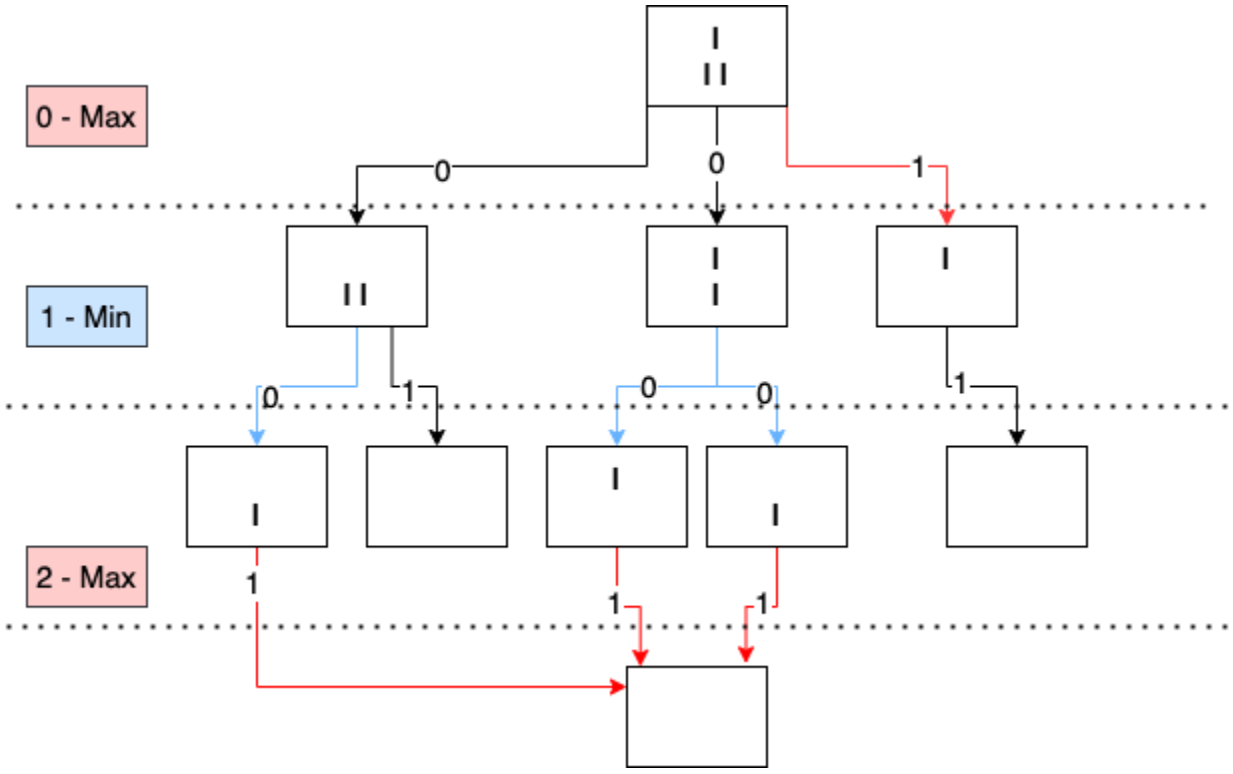
Belief state space is shown above. No solution possible as no path leads to a belief state all of whose elements satisfy the goal.

Q4.

The online search problem can be viewed as an offline search space. The belief state is a set of all possible wall configurations.

- Wall configurations: 2^12 = 4096
- The initial belief state = 2^4096
- The space of the belief state = 9*3^12

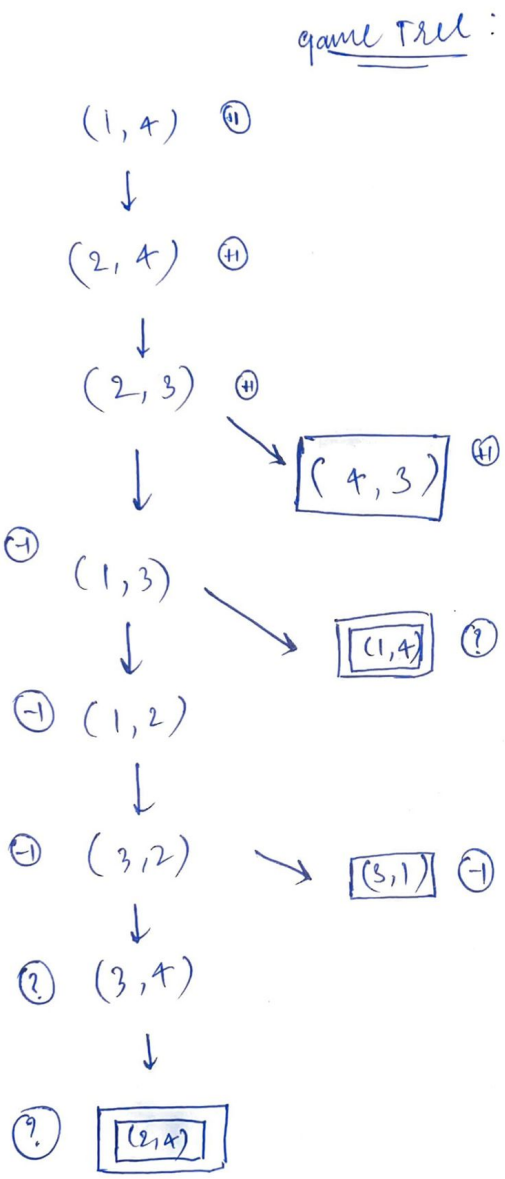
Q5. Game of Nim



- Game of Nim is two player zero-sum game where the last person to pick from the pile loses. There are n sticks in a pile and there are m piles. A player can pick any number of stick from the stack but only has to pick from only one pile.
- Every vertex corresponds to a point at which a decision needs to be made by one player. Each edge emanating from a vertex represents an action.
- The root of the tree indicates the beginning of the game, which usually means that chooses an action. The leaves of the tree represent the end of the game, which are the points at which a cost is received. The cost is usually shown below each leaf.

Q6.

A.



a.

b. It is assumed that when an agent enters ? values, he/she always choses to win, given the situation.

B.

- The evaluation function instead of a single value, is a vector of multiple values. Hence Minimax algorithm works in this case
- Alpha-Beta Pruning does not work as the unvisited leaf nodes might be useful to both the players when there are no constraints on the two terminal utilities.

Q7.

1. Game Tree Construction:
    - The Tree for connect four is a directed graph whose nodes are positions in the game.
    - The edges are the moves where each player drops colored tokens into a seven-column, six-row vertically suspended grid.
2. Connect four is two person zero-sum game where both the players take turns to make a move and know the exact state of the game at all times. Player A changes the state of the problem every step in a direction the other does not want. Each player tries to connect colors and at the same time prevent the other player to do the same in the same grid space. Hence, this problem can be posed as an adversarial search.
3.

```
Evalutaion Funtion(p):
  IF p is a win for MAX(return 9999)
    else
  IF p is a win for MIN(return -9999)
    else
  return val

val <- (possible winning rows, columns, diagonals for MAX)
-(possible winning rows, columns, diagonals for MIN)

Minimax(board):
  best.mv = [not yet defined]
  best.score = -9999
  For each legal move m
  { make move m.mv on Board
    m.score = MIN
    if (m.score > best.score) then best = m
    retract move m.mv on Board
```

```
  }
  Make move best.mv

MAX
if (game over) return EVAL-ENDING
else if (max depth) return EVAL
else
best.score = -9999
for each computer legal move m
  { make move m.mv on Board
  m.score = MIN
  if (m.score > best.score) then best = m
  retract move m.mv on Board
  }
return best.score

MIN
if (game over) return EVAL-ENDING
else if (max depth) return EVAL
else
best.score = 9999
for each human legal move m.mv
{ make move m.mv on Board
m.score = MAX
if (m.score < best.score) then best = m
retract move m.mv on Board
}
return best.score
```

4. Alpha-Beta pruning improves the computational speed and saves memory as it does not generate the entire game tree and produce the utility values for all terminal states.
5. Heuristic : *(possible winning rows, columns, diagonals for MAX) -(possible winning rows, columns, diagonals for MIN)*

PA

Q2. Othello

Board	Board Value
arbitraryBoard8	-6
board1	5

Board	Board Value
endgame	100