Assignment - 01

Name: Amit Sutradhar

ID: 22201054

section: 18

CSE422

Date: 25.03.2025

Part-1

1.

start →	(0,0)	(0,1)	(0,2)	(0, 3)	(0,4)	(0,5)	
	(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	
	(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	
	(3,0)	(3,1)	(3,2)	(3,3)	(3, 4)	(3,5)	
	(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	
	(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	← Goal

using A* algorithm,

$\boxed{(0,0)^{10}}$

$(0,0)^{10}$ $\boxed{(0,1)^{10}\ (1,0)^{10}}$

$(1,0)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,0)^{10}}$

$(2,0)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,1)^{10}\ (3,0)^{10}}$

$(3,0)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,1)^{10}\ (3,1)^{10}\ (4,0)^{10}}$

$(4,0)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,1)^{10}\ (3,1)^{10}\ (4,1)^{10}\ (5,0)^{10}}$

$(5,0)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,1)^{10}\ (3,1)^{10}\ (4,1)^{10}\ (5,1)^{10}}$

$(5,1)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,1)^{10}\ (3,1)^{10}\ (4,1)^{10}\ (5,2)^{10}}$

$(5,2)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,1)^{10}\ (3,1)^{10}\ (4,1)^{10}\ (5,3)^{10}}$

$(5,3)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,1)^{10}\ (3,1)^{10}\ (4,1)^{10}\ (4,2)^{10}\ (4,3)^{10},\ (5,4)^{10}}$

$(5,4)^{10}$ $\boxed{(0,1)^{10}\ (1,1)^{10}\ (2,1)^{10}\ (3,1)^{10}\ (4,1)^{10}\ (4,2)^{10}\ (4,3)^{10}\ (4,4)^{10}\ (5,5)^{10}}$

∵ $(5,5)^{10}$ $\boxed{(0,1)^{10}(1,1)^{10}(2,1)^{10}(3,1)^{10}(4,1)^{10}(4,2)^{10}(4,3)^{10}(4,4)^{10}}$

∴ Path from start to goal:

$(0,0) \rightarrow (1,0) \rightarrow (2,0) \rightarrow (3,0) \rightarrow (4,0) \rightarrow (5,0) \rightarrow (5,1) \rightarrow (5,2)$
$\rightarrow (5,3) \rightarrow (5,4) \rightarrow (5,5)$

2.

If the diagonal moves were allowed:

The heuristic function should change from Manhattan distance to chebyshev distance.

Diagonal moves always allow both x and y at a time, so if chebyshev distance is implemented, it will take the maximum of x and y co-ordinates better estimates the remaining cost.

Part-2

a)

for state representation:

1 = selected

0 = not selected

neighborhood:

All the states can be reached by flipping one bit. Like, adding a coure or removing a course.

Evaluation function:

The work-load per week $\leq$ 50 hours

b) Let,

initial state: No courses selected.

workload = 0 | state : 0
marks = 0

1st Iteration:

lets select course - 10

The state will be = 1

workload = 19, maks = 94

Iteration 2:

let's select course - 5

The state will be - 11

workload = 35, marks = 167

The problems with hill climbing algorithm:

1. This can sometimes lead to suboptimal solution.

2. May get stuck at local maxima.

3. No backtracking

c)

First choice Hill climbing:

Randomly generates neighbors one at a time untill finding one better than the current state. Accept the first improvement found.

Like- current state: [3,6] (15 hours, 134 marks)

add 1 → [1,3,6] (224, 206) → better

[accept immediately]

Stochastic hill climbing:

(i) Evaluates all better neighbors

(ii) Select one using probability and better neighbors have higher selection probability.

Random - Restart hill climbing:

(i). Runs basic hill climbing multiple times from random initial states

(ii) keeps the best solution found across all restarts.

Like,

start → 2 → climbs [2,6] (19h, 112 marks)

start → 5 → climbs [3,5,6] (31h, 207 marks)

↓

~~bet f~~

best found

d) Let,

$T = 100$, $\alpha = 0.5$

Iteration - 1:

current=[5,7] [32 hours, 137 marks]

add course - 10

[5,7,10] . [51 hours, 231 marks]

↓

exceed limit

$\therefore T = 100 \times 0.5^1 = 50$

Iteration - 2:

current = [3,6] [15h, 134 marks]

add course - 5,

[3,5,6] [31h, 207 marks]

$\therefore \Delta E = (207 - 134) = +73$

$\therefore T = 100 \times 0.5^2 = 25$

Iteration - 3:

current [3,5,6,8] [35h, 245 maks]

add course - 10

[3,5,6,8,10] [54 h, 339 maks]

↓

exceed limit

$\therefore T = 100 \times 0.5^3 = 12.5$

e) Initial Population =

1 1 1 1 0001 1 0 = 50 hrs

0 0 1 0 1 1 0 0 1 0 = 35 hrs

0 0 1 1 1 1 0 1 1 0 = 46 hrs

1 1 1 1 0 1 0 1 0 0 = 49 hrs

Crossover:

1 1 1 1 | 0 0 0 1 1 0
0 0 1 0 | 1 1 0 0 1 0

1 1 1 1 1 1 0 0 1 0 → ①
0 0 1 0 0 0 0 1 1 0 → ②

0 0 1 1 | 1 1 0 1 1 0
1 1 1 1 | 0 1 0 1 0 0

0 0 1 1 0 1 0 1 0 0 – ③
1 1 1 1 1 1 0 1 1 0 – ④

mutation offspring - 01

1 0 1 1 1 0 0 0 1 0 → 46 hours

mutation offspring - 02

1 0 1 0 0 0 1 1 1 0 → 43 hours

mutation offspring - 03

0 0 1 0 0 1 0 1 1 0 → 23 hours

mutation offspring - 04

1 0 1 1 0 1 0 1 1 0 → 37 hours

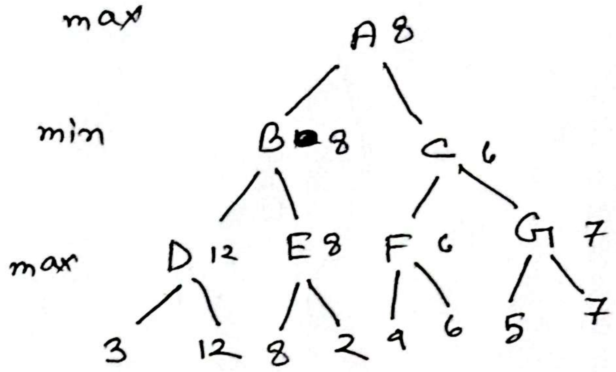
f) If all the chromosomes In the Initial population are same, then no genetic diversity will happen. And algorithm cannot explore new solutions. And also may converge to suboptimal solution when better combinations exist.

Mutation heps to maintain the diversity and can discovers better combinations. It also helps to escape local optima.

—

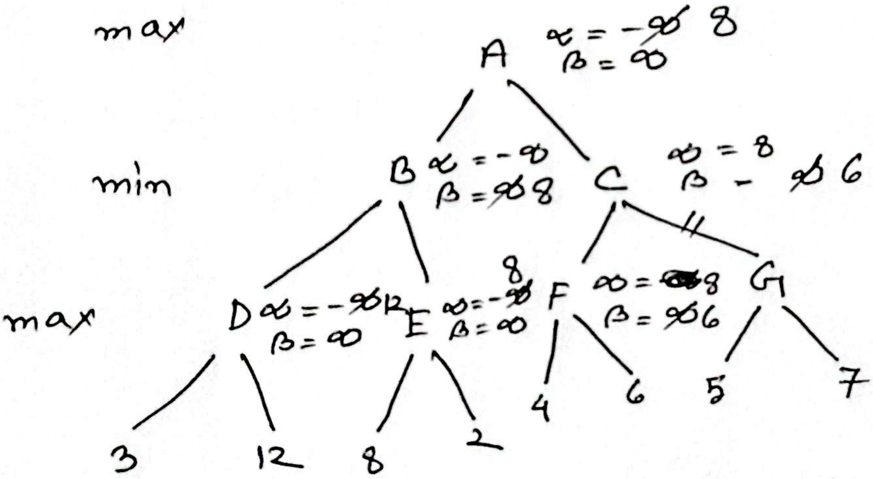
Part - 3

1.

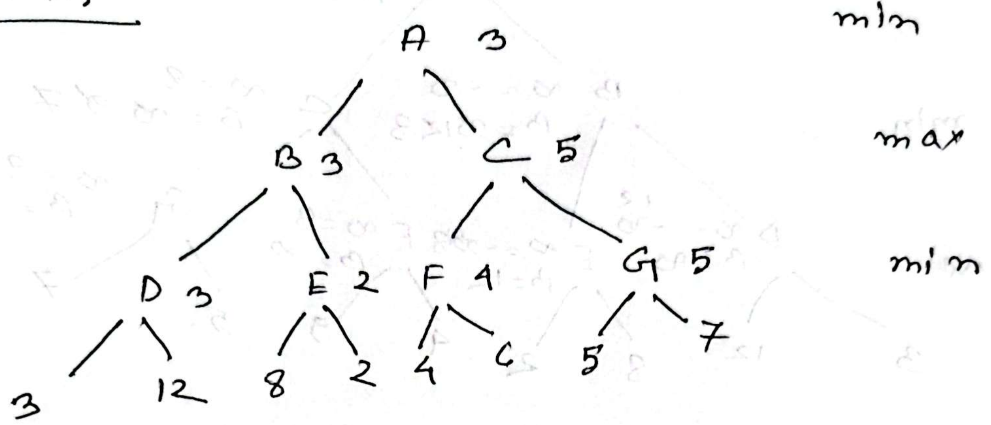


minmax

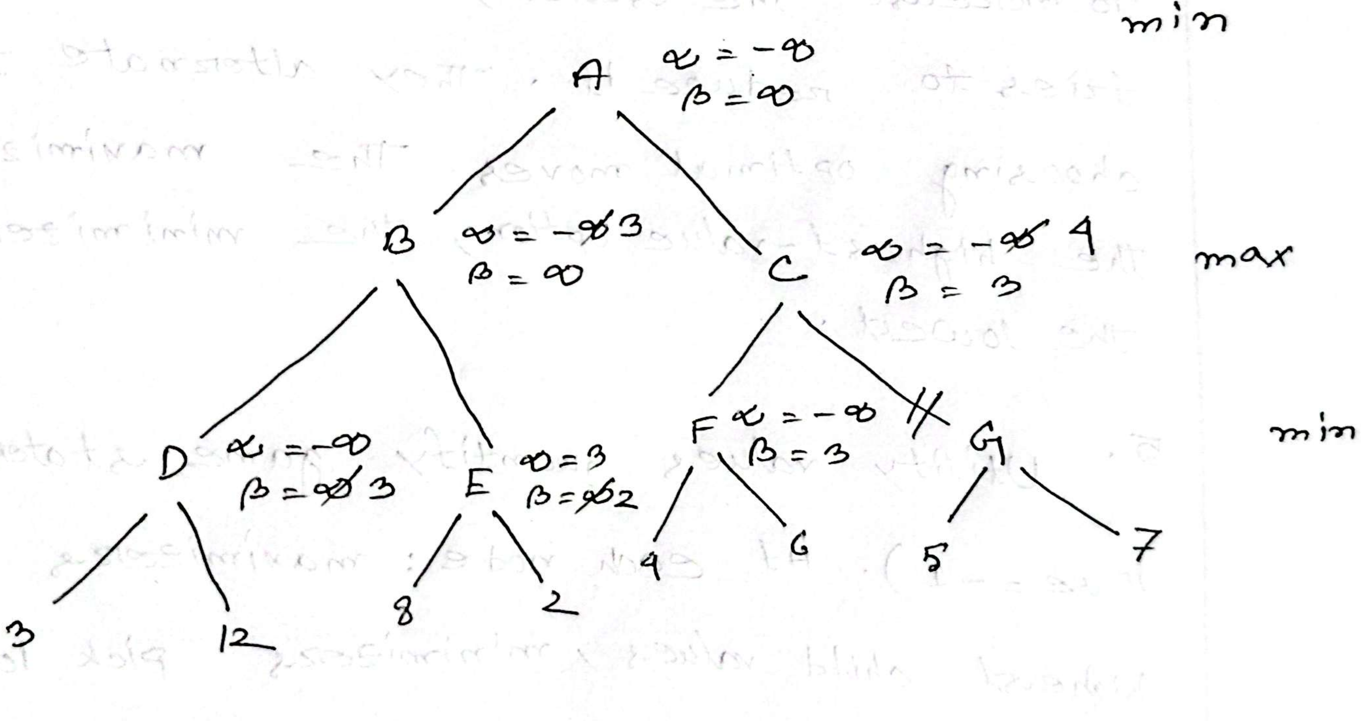
alpha-beta pruning



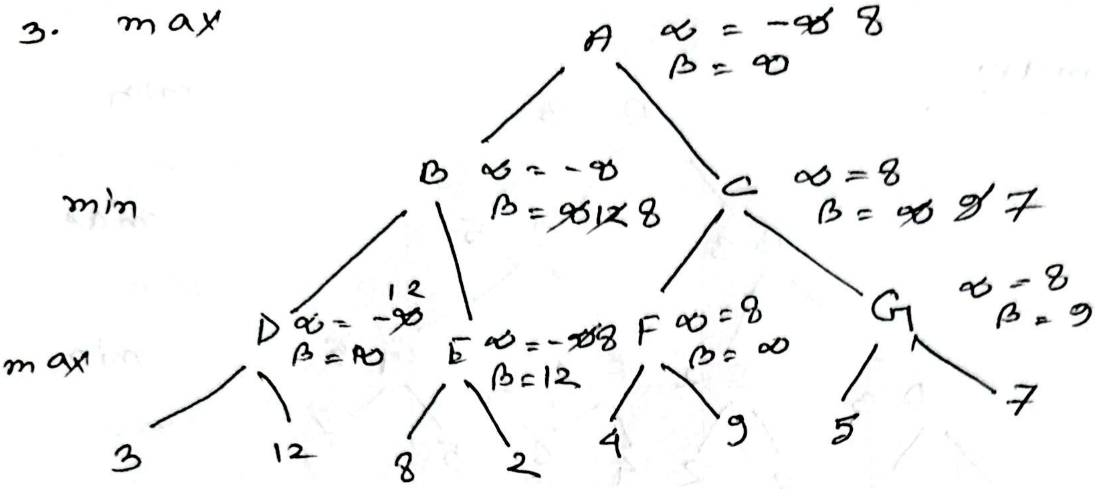
2. minimax:



alpha beta pruning:



3.



4. The maximizing player selects moves to increase the score, while the minimizing player tries to reduce it. They alternate turns, each choosing optimal moves. The maximizer picks the highest-value option, the minimizer selects the lowest.

5. Utility values quantify game states (win = +1, lose = -1). At each node: maximizers choose highest child values, minimizers pick lowest. The utility values are calculated depending on the number of states a game on terminate at.