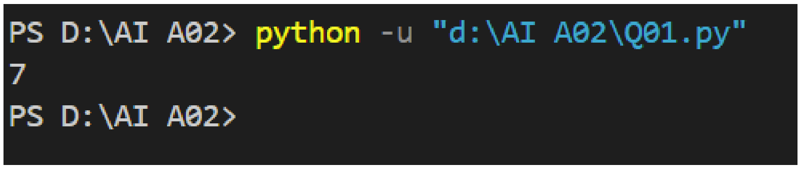
Adina Faraz
23k-0008
BSAI-4A

AI Assignment 02

Question-01 (OUTPUT)



```
PS D:\AI A02> python -u "d:\AI A02\Q01.py"
7
PS D:\AI A02>
```

Question-02

Dry-Run Analysis of Optimizing a Multi-Stage Manufacturing Process Using Genetic Algorithms

Problem Understanding

We need to optimize task allocation across three production facilities while minimizing production time and cost, subject to constraints on facility capacity and task execution costs. A Genetic Algorithm (GA) will be used to achieve this optimization.

Given Parameters

Production Tasks and Times

Task	Time Required (hrs)
Task 1	5
Task 2	8
Task 3	4
Task 4	7
Task 5	6
Task 6	3
Task 7	9

Adina Faraz
23k-0008
BSAI-4A

Production Facilities and Their Capacities

Facility	Capacity (hrs/day)
Facility 1	24
Facility 2	30
Facility 3	28

Task	Facility 1	Facility 2	Facility 3
Task 1	10	12	9
Task 2	15	14	16
Task 3	9	7	7
Task 4	12	10	13
Task 5	14	13	12
Task 6	9	8	10
Task 7	11	12	13

Optimization Goal

Minimize the total production time and costs while ensuring:

1. Each task is assigned to one and only one facility.
2. No facility exceeds its capacity.

Genetic Algorithm Setup

- **Population Size:** 6 chromosomes
- **Crossover Rate:** 80%
- **Mutation Rate:** 20%
- **Selection Method:** Roulette Wheel Selection
- **Crossover Method:** One-Point Crossover
- **Mutation Method:** Swap Mutation
- **Fitness Function:** Based on total cost, penalizing violations of capacity constraints.

Adina Faraz
23k-0008
BSAI-4A

Dry-Run Execution of Genetic Algorithm

Step 1: Encoding the Solution

Each chromosome represents a possible allocation of tasks to facilities. Example chromosome representation:

- `[3, 1, 2, 1, 3, 2, 1]` (Task 1 → Facility 3, Task 2 → Facility 1, etc.)

Step 2: Initial Population Generation

We randomly generate 6 chromosomes:

1. `[1, 2, 3, 1, 2, 3, 1]` (Cost: 481)
2. `[2, 3, 1, 3, 1, 2, 2]` (Cost: 500)
3. `[3, 1, 2, 1, 3, 2, 1]` (Cost: 470)
4. `[1, 3, 2, 2, 1, 1, 3]` (Cost: 495)
5. `[2, 1, 3, 3, 2, 3, 2]` (Cost: 510)
6. `[3, 2, 1, 1, 2, 1, 3]` (Cost: 475)

Step 3: Fitness Function Calculation

We calculate the fitness as the inverse of cost:

Chromosome	Cost	Fitness (1/Cost)	Selection Probability
`[1,2,3,1,2,3,1]`	481	0.00208	17.13%
`[2,3,1,3,1,2,2]`	500	0.00200	16.44%
`[3,1,2,1,3,2,1]`	470	0.00213	17.58%
`[1,3,2,2,1,1,3]`	495	0.00202	16.51%

[2,1,3,3,2,3,2]	510	0.00196	16.07%
[3,2,1,1,2,1,3]	475	0.00211	17.25%

Total probability = 100%

Step 4: Selection (Roulette Wheel Selection)

- We randomly generate a selection point.
- Chromosomes with higher probability are more likely to be chosen.
- The two selected parents based on probability are [3,1,2,1,3,2,1] and [3,2,1,1,2,1,3].

Step 5: Crossover (One-Point Crossover)

- Example parents:
  - Parent 1: [3,1,2,1,3,2,1]
  - Parent 2: [3,2,1,1,2,1,3]
- Crossover at index 4 produces offspring:
  - Offspring 1: [3,1,2,1,2,1,3]
  - Offspring 2: [3,2,1,1,3,2,1]

Step 6: Mutation (Swap Mutation)

- Swap allocations between tasks randomly.
- Example: Mutation swaps Task 2 with Task 5 in [3,1,2,1,2,1,3].
- New offspring: [3,5,2,1,2,1,3].

Step 7: New Generation and Iteration

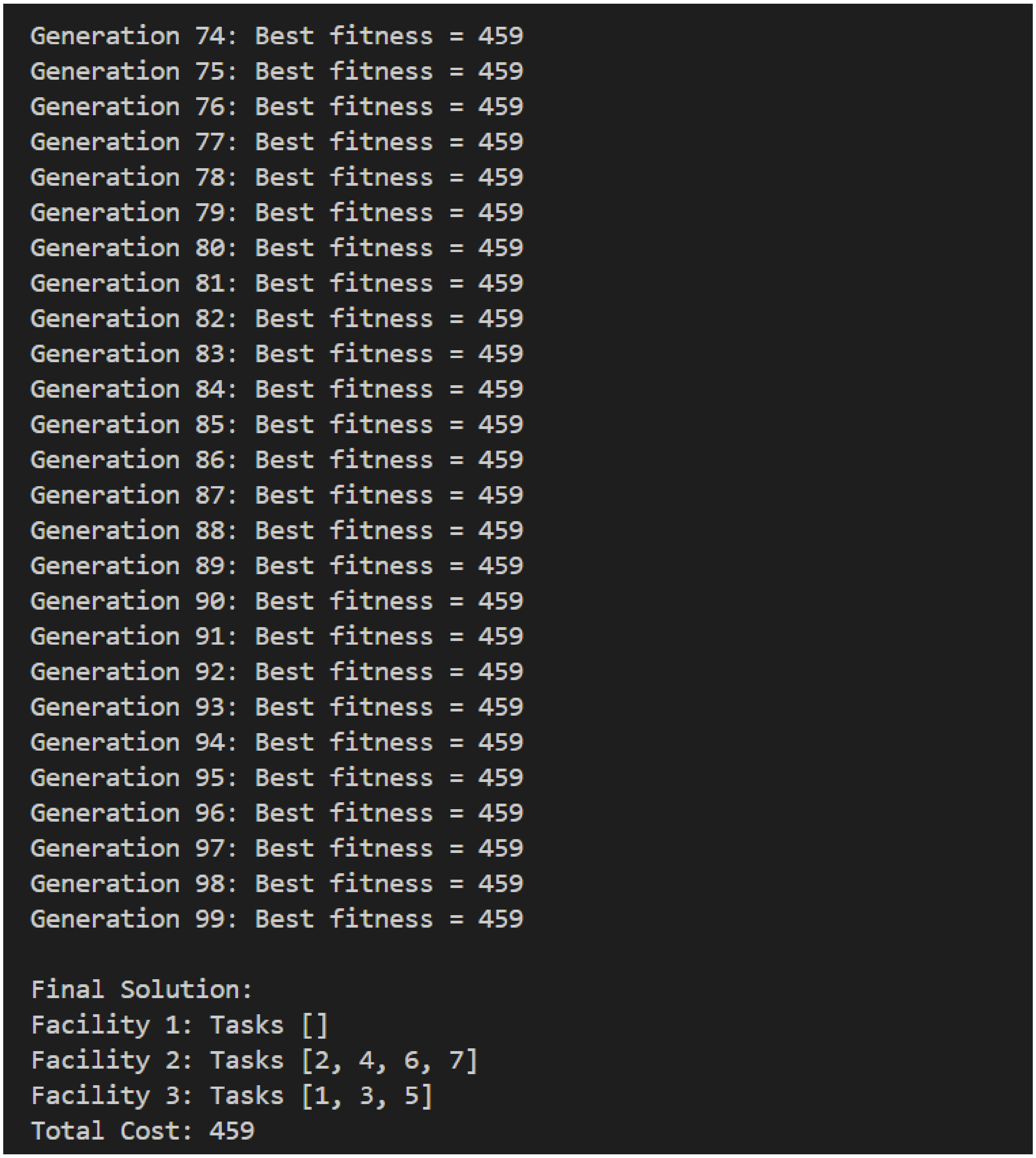
- Evaluate new population fitness.
- Repeat steps 3-6 until convergence.

Final Result Interpretation

- The best chromosome after multiple generations provides the optimal task-to-facility allocation.
- Constraints are met while minimizing costs.
- Final task assignments and total cost are computed.

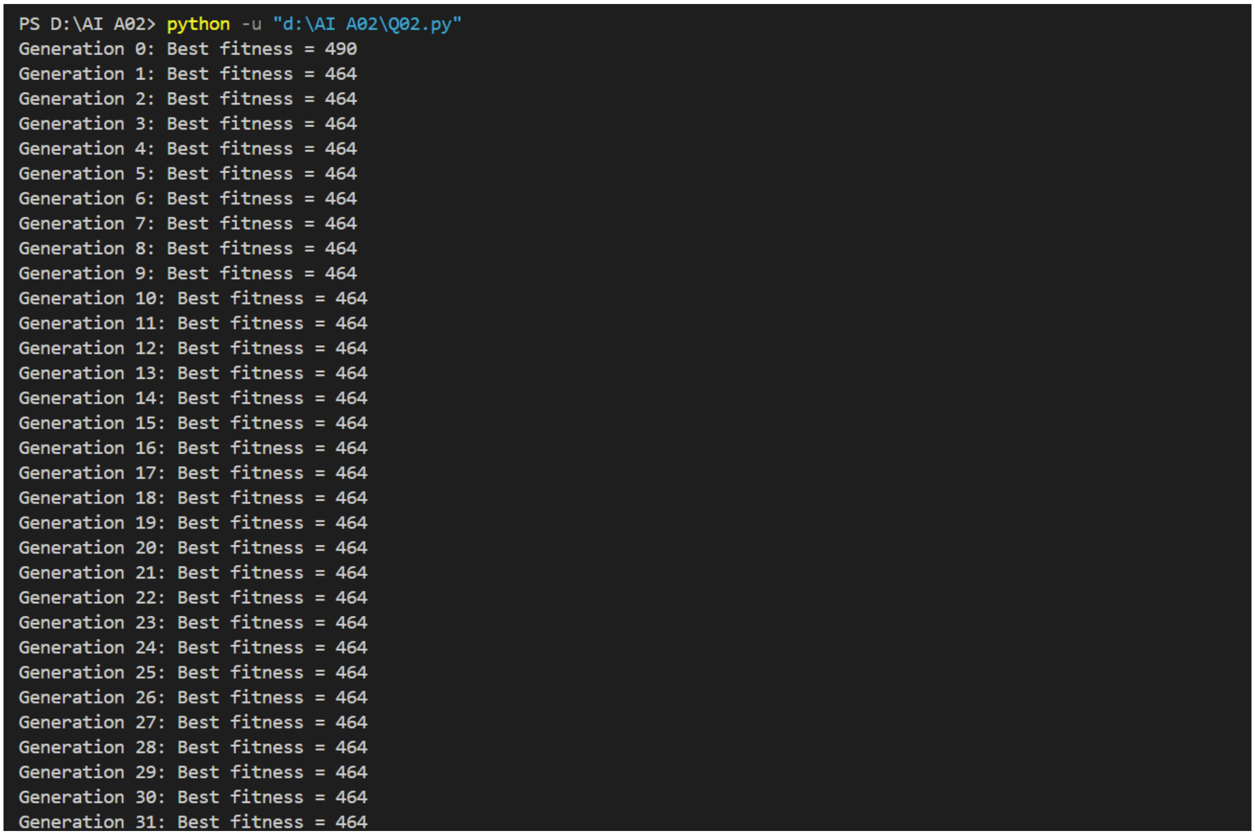
OUTPUT:

Adina Faraz
23k-0008
BSAI-4A



```
Generation 74: Best fitness = 459
Generation 75: Best fitness = 459
Generation 76: Best fitness = 459
Generation 77: Best fitness = 459
Generation 78: Best fitness = 459
Generation 79: Best fitness = 459
Generation 80: Best fitness = 459
Generation 81: Best fitness = 459
Generation 82: Best fitness = 459
Generation 83: Best fitness = 459
Generation 84: Best fitness = 459
Generation 85: Best fitness = 459
Generation 86: Best fitness = 459
Generation 87: Best fitness = 459
Generation 88: Best fitness = 459
Generation 89: Best fitness = 459
Generation 90: Best fitness = 459
Generation 91: Best fitness = 459
Generation 92: Best fitness = 459
Generation 93: Best fitness = 459
Generation 94: Best fitness = 459
Generation 95: Best fitness = 459
Generation 96: Best fitness = 459
Generation 97: Best fitness = 459
Generation 98: Best fitness = 459
Generation 99: Best fitness = 459

Final Solution:
Facility 1: Tasks []
Facility 2: Tasks [2, 4, 6, 7]
Facility 3: Tasks [1, 3, 5]
Total Cost: 459
```

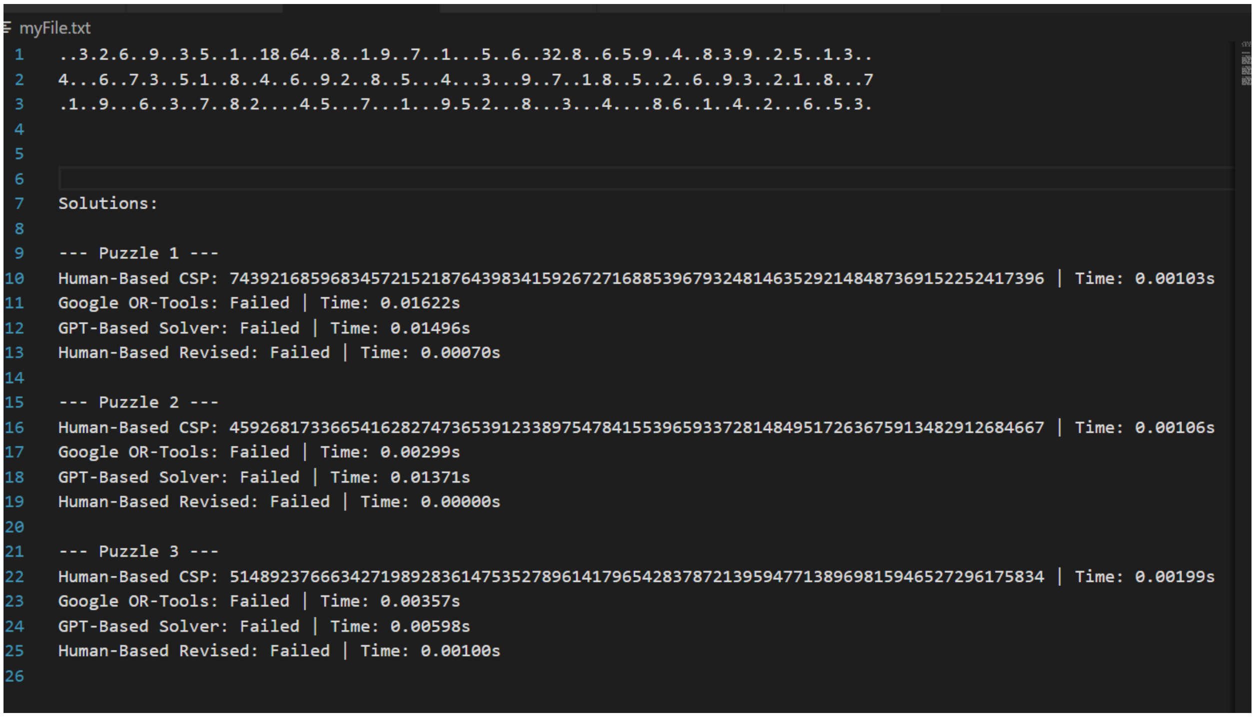
Adina Faraz
23k-0008
BSAI-4A



```
PS D:\AI A02> python -u "d:\AI A02\Q02.py"
Generation 0: Best fitness = 490
Generation 1: Best fitness = 464
Generation 2: Best fitness = 464
Generation 3: Best fitness = 464
Generation 4: Best fitness = 464
Generation 5: Best fitness = 464
Generation 6: Best fitness = 464
Generation 7: Best fitness = 464
Generation 8: Best fitness = 464
Generation 9: Best fitness = 464
Generation 10: Best fitness = 464
Generation 11: Best fitness = 464
Generation 12: Best fitness = 464
Generation 13: Best fitness = 464
Generation 14: Best fitness = 464
Generation 15: Best fitness = 464
Generation 16: Best fitness = 464
Generation 17: Best fitness = 464
Generation 18: Best fitness = 464
Generation 19: Best fitness = 464
Generation 20: Best fitness = 464
Generation 21: Best fitness = 464
Generation 22: Best fitness = 464
Generation 23: Best fitness = 464
Generation 24: Best fitness = 464
Generation 25: Best fitness = 464
Generation 26: Best fitness = 464
Generation 27: Best fitness = 464
Generation 28: Best fitness = 464
Generation 29: Best fitness = 464
Generation 30: Best fitness = 464
Generation 31: Best fitness = 464
```

Question-03 (OUTPUT)

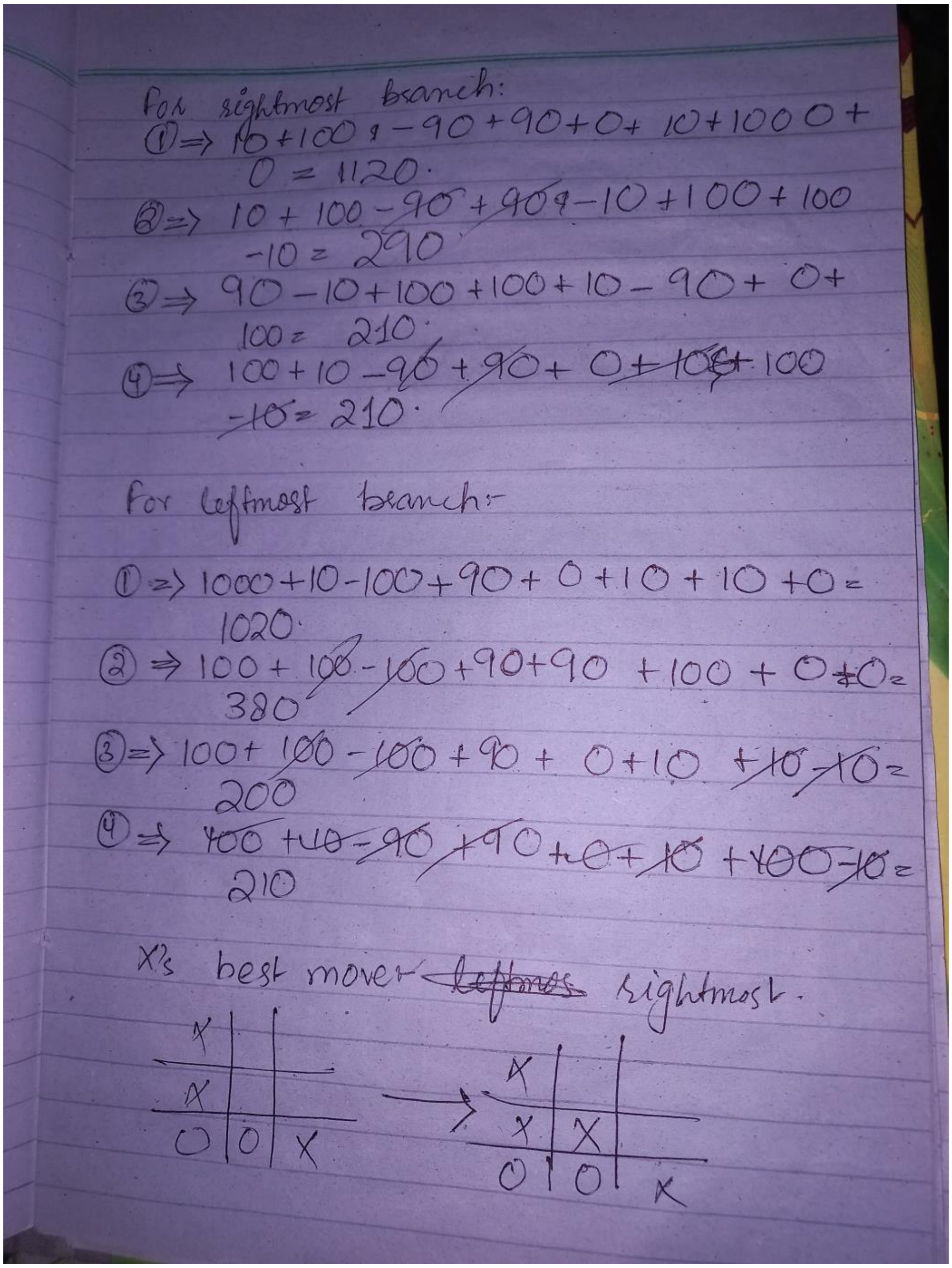
Adina Faraz
23k-0008
BSAI-4A



```
..3.2.6..9..3.5..1..18.64..8..1.9..7..1...5..6..32.8..6.5.9..4..8.3.9..2.5..1.3..
4...6..7.3..5.1..8..4..6..9.2..8..5...4...3...9..7..1.8..5..2..6..9.3..2.1..8...7
.1..9...6..3..7..8.2....4.5...7...1...9.5.2...8...3...4....8.6..1..4..2...6..5.3.



Solutions:

--- Puzzle 1 ---
Human-Based CSP: 743921685968345721521876439834159267271688539679324814635292148487369152252417396 | Time: 0.00103s
Google OR-Tools: Failed | Time: 0.01622s
GPT-Based Solver: Failed | Time: 0.01496s
Human-Based Revised: Failed | Time: 0.00070s

--- Puzzle 2 ---
Human-Based CSP: 459268173366541628274736539123389754784155396593372814849517263675913482912684667 | Time: 0.00106s
Google OR-Tools: Failed | Time: 0.00299s
GPT-Based Solver: Failed | Time: 0.01371s
Human-Based Revised: Failed | Time: 0.00000s

--- Puzzle 3 ---
Human-Based CSP: 514892376663427198928361475352789614179654283787213959477138969815946527296175834 | Time: 0.00199s
Google OR-Tools: Failed | Time: 0.00357s
GPT-Based Solver: Failed | Time: 0.00598s
Human-Based Revised: Failed | Time: 0.00100s

```

Question-04

Adina Faraz
23k-0008
BSAI-4A



For rightmost branch:

① ⇒ $10+100+-90+90+0+10+1000+0 = 1120.$

② ⇒ $10+100-\cancel{90+90}+-10+100+100-10 = 290$

③ ⇒ $90-10+100+100+10-90+0+100 = 210.$

④ ⇒ $100+10\cancel{-90+90}+0\cancel{+10}+100\cancel{-10} = 210.$

For leftmost branch:-

① ⇒ $1000+10-100+90+0+10+10+0 = 1020.$

② ⇒ $100+\cancel{100-100}+90+90+100+0+0 = 380$

③ ⇒ $100+\cancel{100-100}+90+0+10\cancel{+10-10} = 200$

④ ⇒ $\cancel{400+10-90}\cancel{+90}+0\cancel{+10}+400\cancel{-10} = 210$

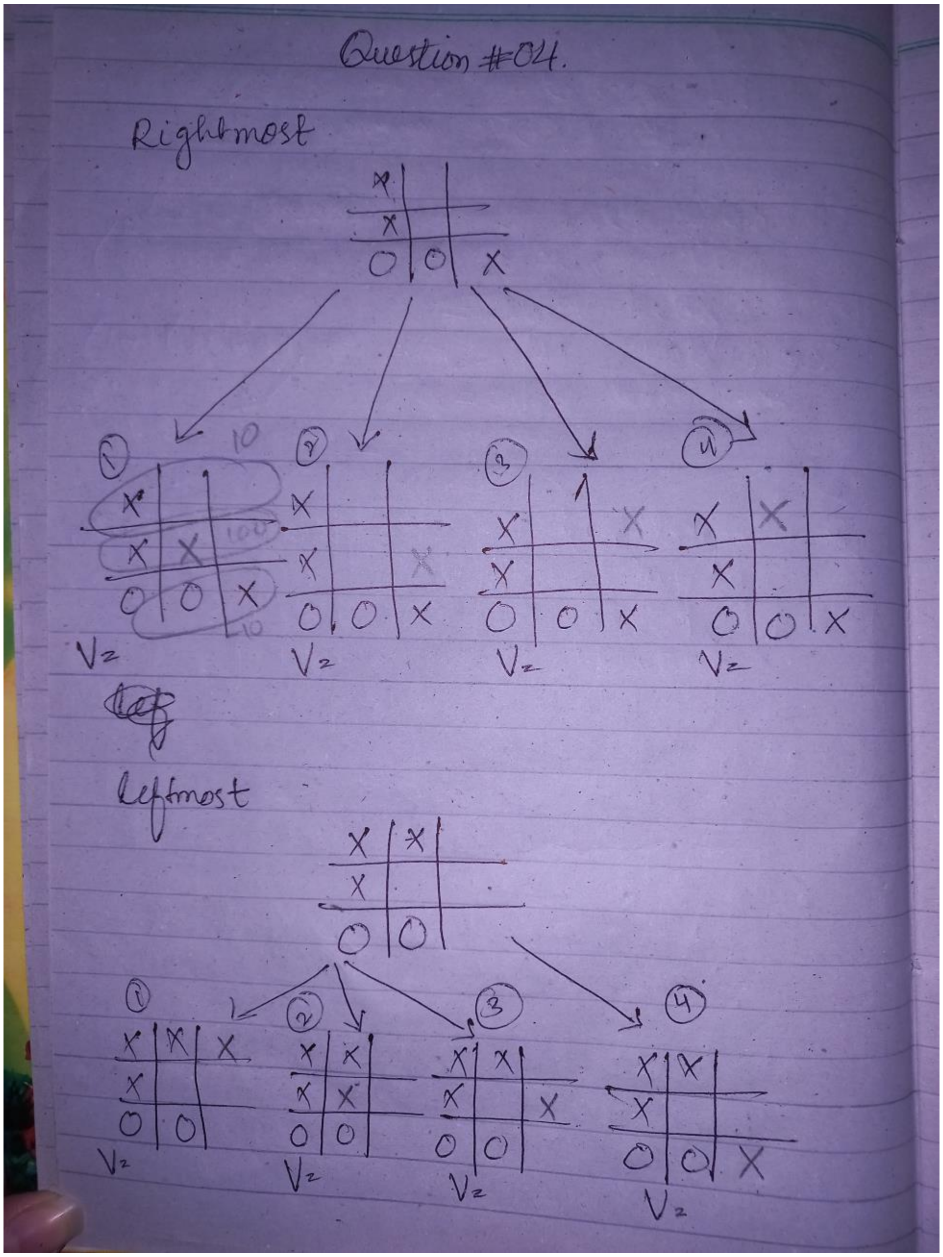
X's best move:- ~~leftmost~~ rightmost.

X		
X		
O	O	X

→

X		
X	X	
O	O	X

Adina Faraz
23k-0008
BSAI-4A



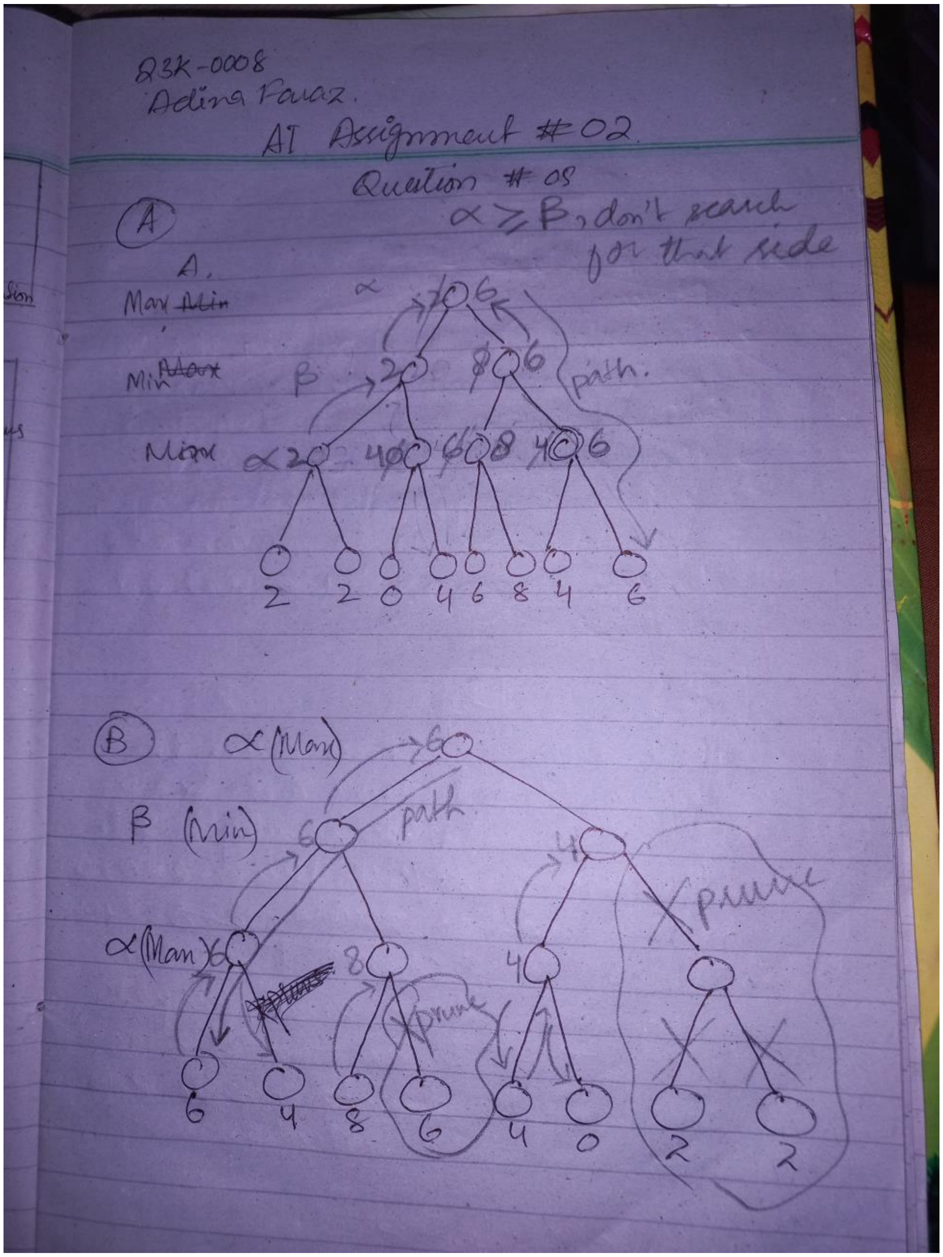
Question #04.

Rightmost

Leftmost

Question-05

Adina Faraz
23k-0008
BSAI-4A



23K-0008
Adina Faraz.

AI Assignment #02

Question # 05

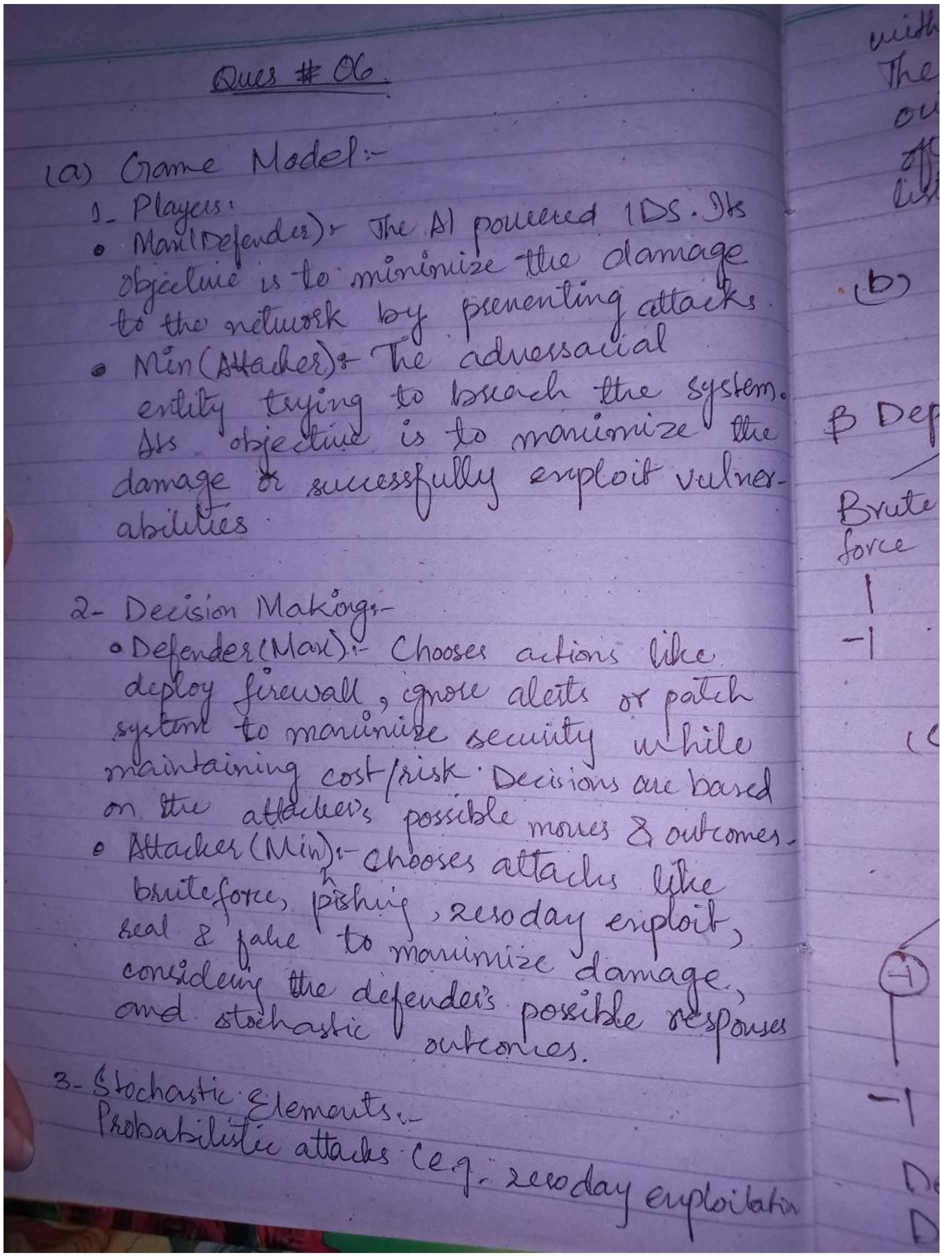
(A)

α ≥ β, don't search for that side

(B)

Question-06

Adina Faraz
23k-0008
BSAI-4A



Ques # 06.

(a) Game Model:-

1- Players:

- Max(Defender):- The AI powered IDS. Its objective is to minimize the damage to the network by preventing attacks.
- Min(Attacker):- The adversarial entity trying to breach the system. Its objective is to maximize the damage or successfully exploit vulnerabilities.

2- Decision Making:-

- Defender(Max):- Chooses actions like deploy firewall, ignore alerts or patch system to maximize security while maintaining cost/risk. Decisions are based on the attacker's possible moves & outcomes.
- Attacker(Min):- Chooses attacks like bruteforce, phishing, zeroday exploit, seal & fake to maximize damage, considering the defender's possible responses and stochastic outcomes.

3- Stochastic Elements:-
Probabilistic attacks (e.g. zeroday exploitation

with
The
ou
off
lik

(b)

B Dep

Brute
force
|
-1

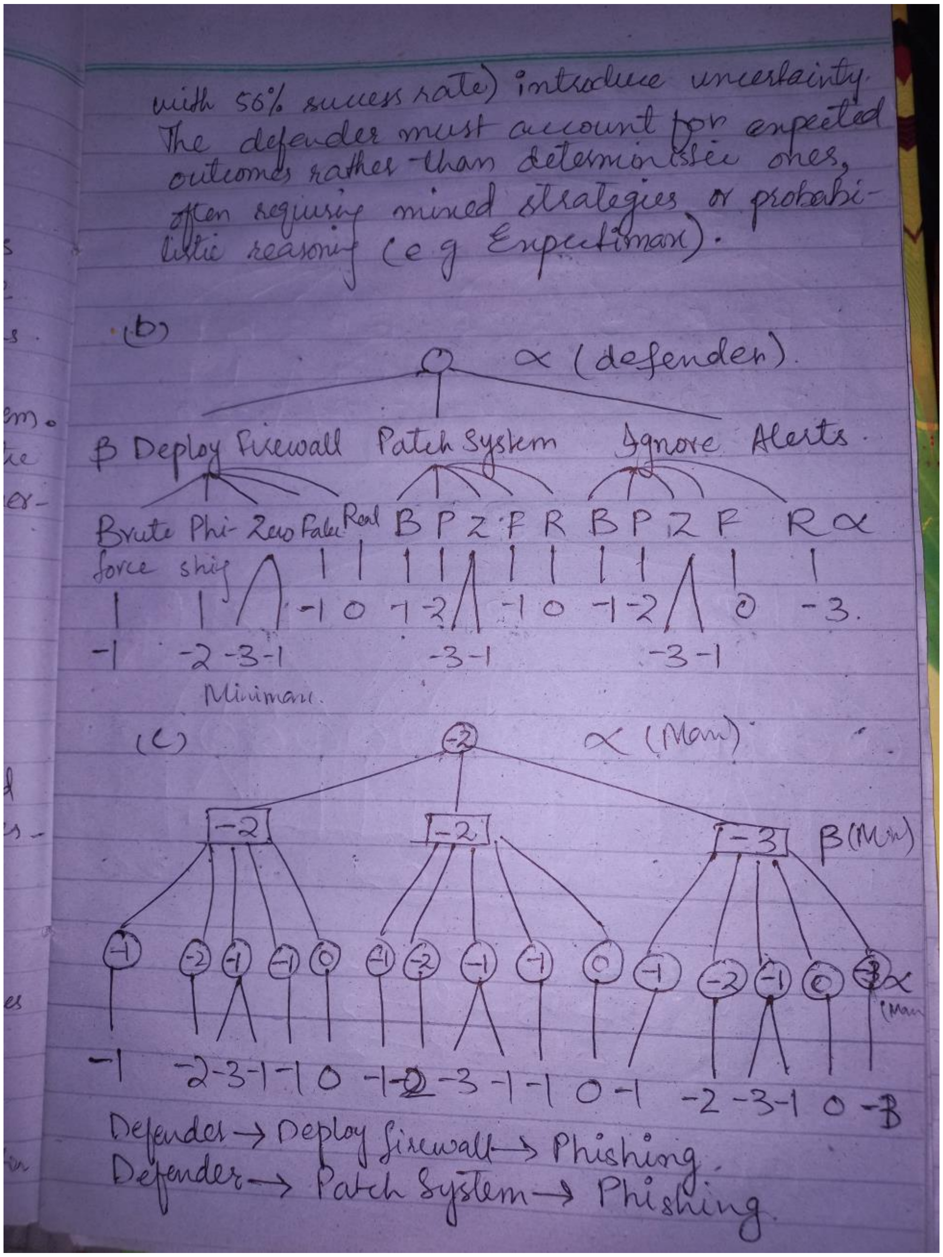
(C

-1

-1

De
D

Adina Faraz
23k-0008
BSAI-4A



with 56% success rate) introduce uncertainty. The defender must account for expected outcomes rather than deterministic ones, often requiring mixed strategies or probabilistic reasoning (e.g Expectimax).

(b)

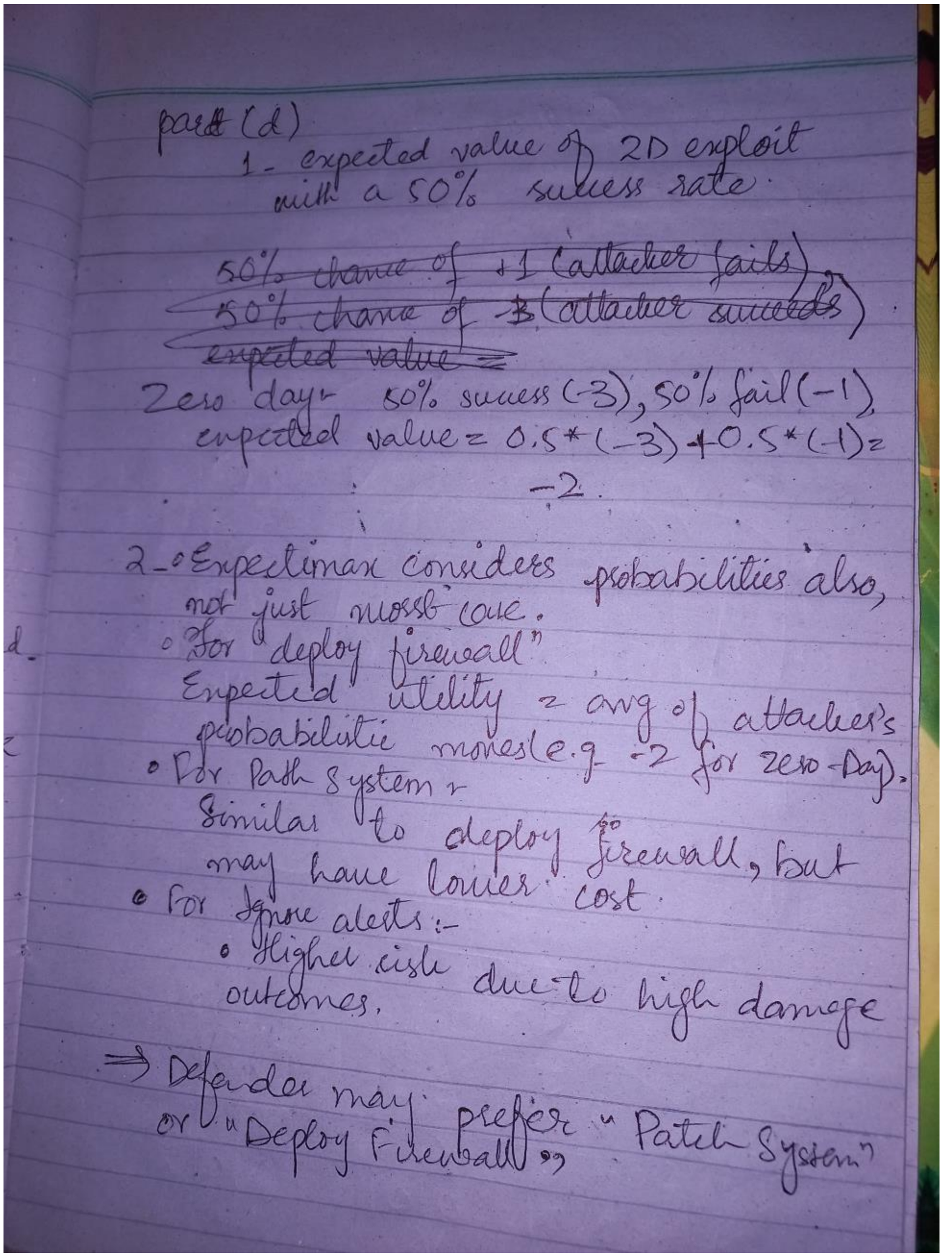
α (defender).
β Deploy Firewall
Patch System
Ignore Alerts.
Brute force
Phi-shing
Zero
Fake
Real
B P Z F R B P Z F R α
-1 -2 -3 -1 -1 0 -1 -2 -3 -1 -1 0 -1 -2 -3 -1 0 -3.
Minimax.

(c)

-2
α (Max)
-2
-2
-3
β (Min)
-1 -2 -1 -1 0 -1 -2 -1 -1 0 -1 -2 -1 0 -3
α (Max)
-1 -2 -3 -1 -1 0 -1 -2 -3 -1 -1 0 -1 -2 -3 -1 0 -3

Defender → Deploy firewall → Phishing.
Defender → Patch System → Phishing.

Adina Faraz
23k-0008
BSAI-4A



part (d)

1- expected value of ZD exploit with a 50% success rate.

~~50% chance of +1 (attacker fails),~~
~~50% chance of -3 (attacker succeeds)~~.
~~expected value =~~

Zero day:- 50% success (-3), 50% fail (-1),
expected value = 0.5*(-3)+0.5*(-1)=
-2.

2- • Expectimax considers probabilities also, not just worst case.

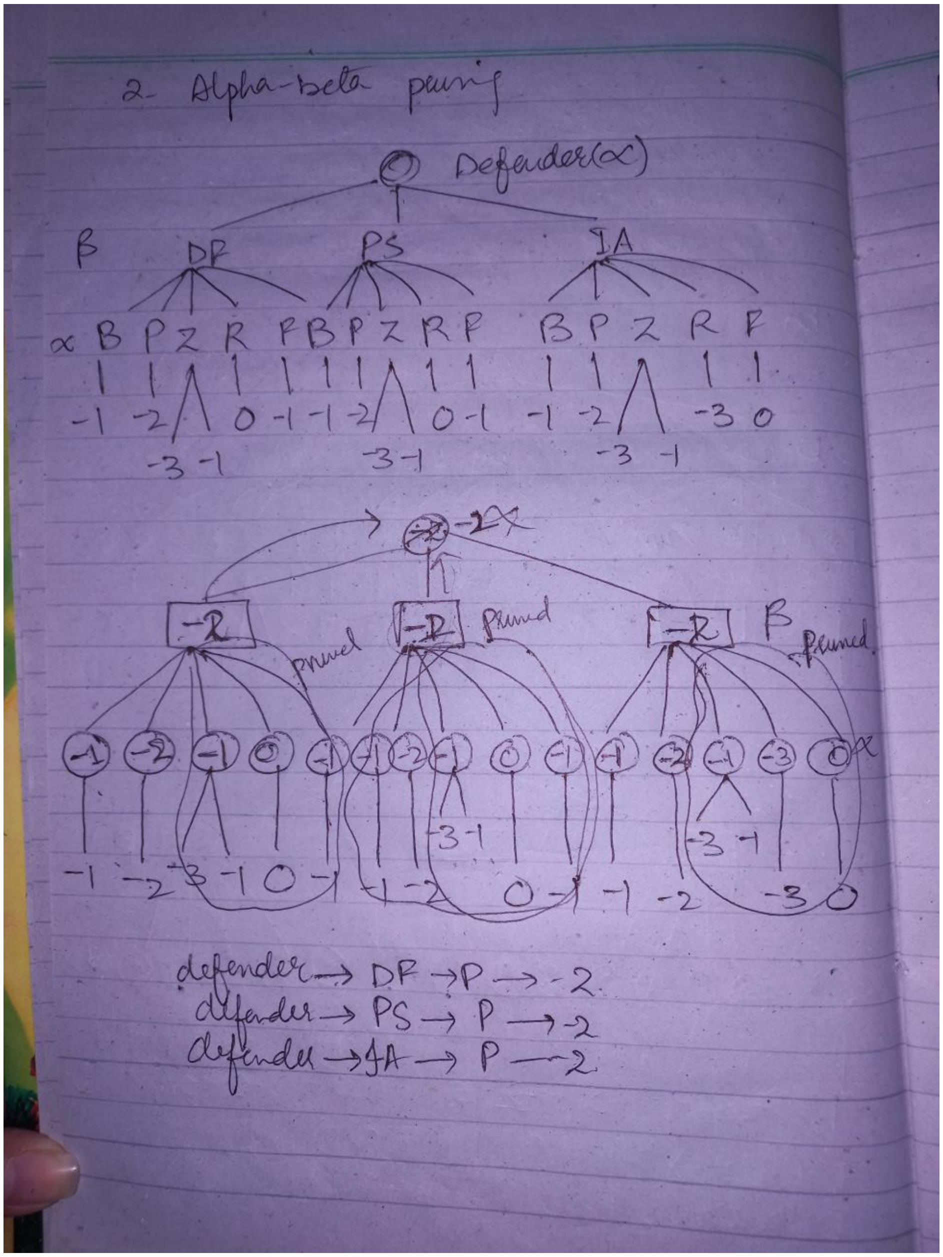
• For "deploy firewall"
Expected utility = avg of attacker's probabilistic moves(e.g. -2 for zero-Day).

• For Path System:-
Similar to deploy firewall, but may have lower cost.

• For Ignore alerts:-
• Higher risk due to high damage outcomes.

→ Defender may prefer "Patch System" or "Deploy Firewall".

Adina Faraz
23k-0008
BSAI-4A



2. Alpha-beta pruing

defender → DF → P → -2
dlfender → PS → P → -2
defendu → IA → P → -2