we asked for a delay in submission and were approved until 1.1

The nodes at the game search algorithms:

We represented a state node as a class with 5 attributes: (list of people, list of broken vertices, scores of each player, location of each agent and turn of which player).

- List of people = list that represent which node has people in it (for example, if v0 has 5 people, v1 has no people and v2 has 3 people, the list should be [5,0,3]).
- List of broken nodes: list that represent which node is broken (for example, if v0 is broken, v1 is broken and v2 isn't broken, the list should be [True, True, False]).
- Scores of each player = represents as a tuple (a, b) where a and b are the amount of people saved by agents A and B, respectively.
- Location of each player = represents as a tuple (a, b) where a and b are the location of agents A and B, respectively.
- True or false

Heuristic function:

We made a simple heuristic function for each game (since it doesn't matter if the heuristic is admissible).

- Adversarial game: the heuristic is the difference between the score of the Max agent and the score of Min agent (A score – B score), since it probably isn't a goal state, its kind of approximate of the score for the Max agent.
- cooperative games: the heuristic is the sum between the score of the Max agent and the score of Min agent (A score + B score), since it probably isn't a goal state, it's kind of approximate of the score for both agent
- semi cooperative game: the heuristic is to choose the highest score and if it's the same choose the other players score

The flow of the program:

- we ask from the user the game tactic, the cutoff parameter, and the starting location for each agent
- each agent searches the best move using the search tree (each tactic has the same search tree, but the score of the leaf is different at each tactic.
- When we finish with the search, we found a leaf node which is the target for the agent that searched. We check recursively what is the first action that the agent did to reach eventually to that node. Then the agent actually performs that action (Traverse or No-Op), and finish his turn.

We use the following graph when agent A(0) start at vertex 0(1) and agent B(1) starts at vertex 1(2) and cuttof 10

We can see that in each run the agent scores and moves are different to reflect the gametype

Please note the the printed verticies number is -1 what they are at the input

```
#N 7
#V1
#V2
#V3 P1
#V4 B
#V5 B
#V6 P2
#V7 P2
#E1 1 4 W1
#E2 1 5 W1
#E3 2 3 W1
#E4 2 5 W1
#E5 3 4 W1
#E6 4 5 W1
#E7 4 6 W1
#E8 5 7 W1
```

Adversarial agent run:

```
C:\Ai-assigment2\venv\Scripts\python.exe C:\Ai-assigment2\main.py

game type? 1 for adv 2 for semi 3 for full

1

depth of cuttof for agents root is 0

10

choose start vertex

0

choose start vertex

1

people=[0, 0, 1, 0, 0, 2, 2], broken=[False, False, False, False, False, False, False] location=[V0, V1] agent.id=0:


agent 0 has moved from 0 to 4


people=[0, 0, 1, 0, 0, 2, 2], broken=[False, False, False, False, True, False, False] location=[V4, V1] agent.id=1:


agent 1 has moved from 1 to 2
```

```
people=[0, 0, 0, 0, 0, 2, 2], broken=[False, False, False, False, True, False, False] location=[V4, V2] agent.id=0:

agent 0 has moved from 4 to 3

people=[0, 0, 0, 0, 0, 2, 2], broken=[False, False, False, True, True, False, False] location=[V3, V2] agent.id=1:

agent 1 has moved from 2 to 1

people=[0, 0, 0, 0, 0, 2, 2], broken=[False, False, False, True, True, False, False] location=[V3, V1] agent.id=0:

agent 0 has moved from 3 to 5

people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, True, False, False] location=[V5, V1] agent.id=1:

agent 1 has moved from 1 to 2

people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, True, False, False] location=[V5, V2] agent.id=0:

tried to move to myself probobly tried to give the other agent a good move

agent 0 has moved from 5 to 5

people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, True, False, False] location=[V5, V2] agent.id=1:

agent 1 has moved from 2 to 1

people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, True, False, False] location=[V5, V1] agent.id=0:

tried to move to myself probobly tried to give the other agent a good move

agent 0 has moved from 5 to 5

people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, True, False, False] location=[V5, V1] agent.id=1:
```

```
game has ended

AdversarialAgent 0 has been removed with a score of 1995.000000 saved 2 with the time of 5

AdversarialAgent 1 has been removed with a score of 996.000000 saved 1 with the time of 4


Process finished with exit code 0
```

```
Semi agent run: C:\Ai-assigment2\venv\Scripts\python.exe C:\Ai-assigment2\main.py
game type? 1 for adv 2 for semi 3 for full
2
depth of cuttof for agents root is 0
10
choose start vertex
0
choose start vertex
1
people=[0, 0, 1, 0, 0, 2, 2], broken=[False, False, False, False, False, False, False] location=[V0, V1] agent.id=0:

agent 0 has moved from 0 to 3

people=[0, 0, 1, 0, 0, 2, 2], broken=[False, False, False, True, False, False, False] location=[V3, V1] agent.id=1:

agent 1 has moved from 1 to 4

people=[0, 0, 1, 0, 0, 2, 2], broken=[False, False, False, True, True, False, False] location=[V3, V4] agent.id=0:

agent 0 has moved from 3 to 5

people=[0, 0, 1, 0, 0, 0, 2], broken=[False, False, False, True, True, False, False] location=[V5, V4] agent.id=1:

agent 1 has moved from 4 to 6

people=[0, 0, 1, 0, 0, 0, 0], broken=[False, False, False, True, True, False, False] location=[V5, V6] agent.id=0:

tried to move to myself probobly tried to give the other agent a good move

agent 0 has moved from 5 to 5

people=[0, 0, 1, 0, 0, 0, 0], broken=[False, False, False, True, True, False, False] location=[V5, V6] agent.id=1:
```

```
tried to move to myself probobly tried to give the other agent a good move

agent 1 has moved from 6 to 6

people=[0, 0, 1, 0, 0, 0, 0], broken=[False, False, False, True, True, False, False] location=[V5, V6] agent.id=0:

game has ended

SemiCooperativeAgent 0 has been removed with a score of 1997.000000 saved 2 with the time of 3

SemiCooperativeAgent 1 has been removed with a score of 1997.000000 saved 2 with the time of 3


Process finished with exit code 0
```

Fully agent run:

```
C:\Ai-assigment2\venv\Scripts\python.exe C:\Ai-assigment2\main.py
```

```
game type? 1 for adv 2 for semi 3 for full

3

depth of cuttof for agents root is 0

10

choose start vertex

0

choose start vertex

1

people=[0, 0, 1, 0, 0, 2, 2], broken=[False, False, False, False, False, False, False] location=[V0, V1] agent.id=0:


agent 0 has moved from 0 to 3


people=[0, 0, 1, 0, 0, 2, 2], broken=[False, False, False, True, False, False, False] location=[V3, V1] agent.id=1:


agent 1 has moved from 1 to 2


people=[0, 0, 0, 0, 0, 2, 2], broken=[False, False, False, True, False, False, False] location=[V3, V2] agent.id=0:


agent 0 has moved from 3 to 5


people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, False, False, False] location=[V5, V2] agent.id=1:


agent 1 has moved from 2 to 1


people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, False, False, False] location=[V5, V1] agent.id=0:


tried to move to myself probobly tried to give the other agent a good move


agent 0 has moved from 5 to 5


people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, False, False, False] location=[V5, V1] agent.id=1:


agent 1 has moved from 1 to 4
```

```
people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, True, False, False] location=[V5, V4] agent.id=0:

tried to move to myself probobly tried to give the other agent a good move

agent 0 has moved from 5 to 5

people=[0, 0, 0, 0, 0, 0, 2], broken=[False, False, False, True, True, False, False] location=[V5, V4] agent.id=1:

agent 1 has moved from 4 to 6

people=[0, 0, 0, 0, 0, 0, 0], broken=[False, False, False, True, True, False, False] location=[V5, V6] agent.id=0:

game has ended

FullyCooperativeAgent 0 has been removed with a score of 1996.000000 saved 2 with the time of 4

FullyCooperativeAgent 1 has been removed with a score of 2996.000000 saved 3 with the time of 4



Process finished with exit code 0
```