

Advanced Data Structure and Algorithms Mini-Problem



Important information

To do this mi-project:

- Make groups of only two students.
- Make report on the theoretical study, modification, diagrams or schemas and discussions on the results.
- Do not include any code in this theoretical report. Please make attention to the written quality and format of your report.
- For each step, please detail the model, the algorithm "no code" and the discussion.
- All parts must be coded in a "python or C #" project. One language for all the projects. All part must be coded in the project.
- Create clean, functional and commented code and display the results.

The deliverables

- PDF report between 10 to 20 pages
- Zip of code

The mini-problem is also available online: <https://complex-systems-ai.com/algorith/adsa-project-among-us/>

Step 1: To organize the tournament

Zerator asks you to organize the next “Among Us” tournament for the next ZLAN.

The rules are as following:

- Total of 100 players
- 10 players per game
- 3 random games then
- each game regroups the players by a batch of ten following their ranking.
 - The last 10 players (in the ranking) are ejected to the tournament.
 - Do it until it remains only 10 players
- For the last 10 players, play 5 games with reinitiated ranking. Update and check the ranking of the 10 players and give the podium.

Here is the ranking model:

- Impostor: 1pts per kill, 3pts per undiscovered murder, 10pts if win
- Crewmate: 3pts if the argument unmasks an imposter, 1pts if all solo tasks are made, 5pts if win

Each time a game ended, the score of each player is the mean of all its games.

The players are stored in a structured database with a log complexity to reach an element which corresponds to a score (the most optimized structure presented in the ADSA Course).

Argue about the question, present the code and display the results.

Argue about the question, present the code and display the results.

1. Propose a data structure to represent a Player and its Score
2. Propose a most optimized data structures for the tournament (called database in the following questions)
3. Present and argue about a method that randomize player score at each game (between 0 point to 12 points)
4. Present and argue about a method to update Players score and the database
5. Present and argue about a method to create random games based on the database

6. Present and argue about a method to create games based on ranking
7. Present and argue about a method to drop the players and to play game until the last 10 players
8. Present and argue about a method which display the TOP10 players and the podium after the final game.

Step 2: Professor Layton < Guybrush Threepwood < You



You're not only the tournament organizer, you're also a player. Thus, you must find the best strategies to grab points to climb the ladder.

Most of the time, the two Impostors are among Crewmates. They never walk together. Thus, the information about players which are seen together may help to find the Impostor. After the first kill's report, the following information presents the players which see each other's:

- Player 0 has seen player 1, 4 and 5
- Player 1 has seen player 0, 2 and 6
- Player 2 has seen player 1, 3 and 7
- Player 3 has seen player 2, 4 and 8
- Player 4 has seen player 0, 3 and 9
- Player 5 has seen player 0, 7 and 8
- Player 6 has seen player 1, 8 and 9
- Player 7 has seen player 2, 5 and 9

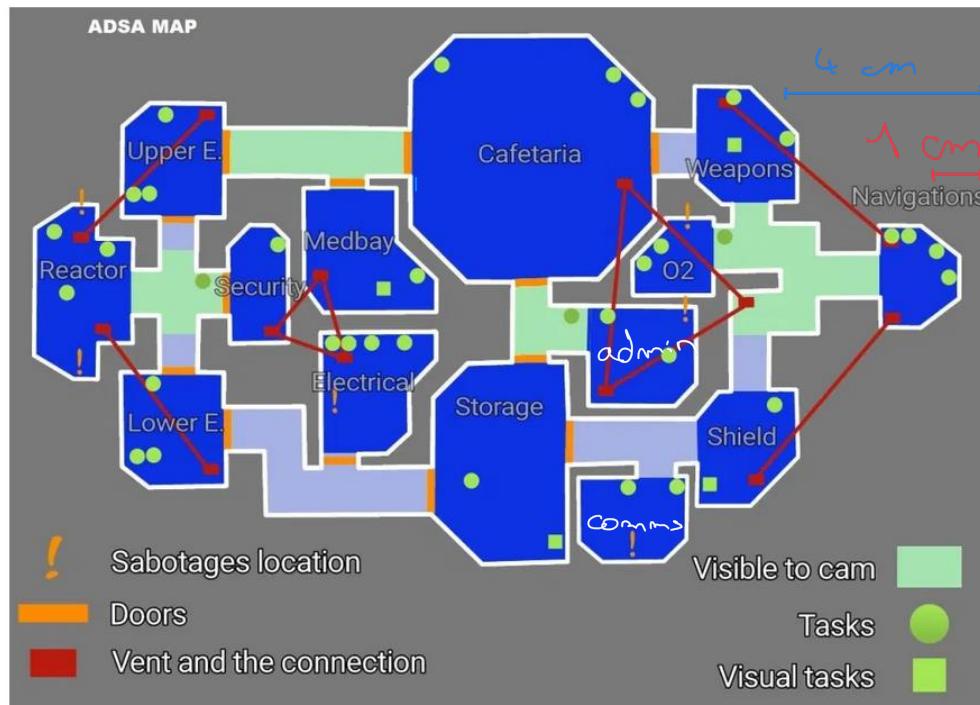
- Player 8 has seen player 3, 5 and 6
- Player 9 has seen player 4, 6 and 7.

Player 0 has been reported dead. So, 1, 4 and 5 may be an impostor. Considering the second impostor hasn't seen player 1, 4 or 5, define a set of probable impostors.

Argue about the question, present the code and display the results.

1. Represent the relation (have seen) between players as a graph, argue about your model.
2. Thanks to a graph theory problem, present how to find a set of probable impostors.
3. Argue about an algorithm solving your problem.
4. Implement the algorithm and show a solution.

Step 3: I don't see him, but I can give proofs he vents!



All the game will be on [this map.](#)

Considering that a player (a crewmate) can only walk through the map, but an impostor can also travel with vent, it is important to compute the time to travel between each room for crewmates and impostors.

A room is represented by its center (you don't have to be precise). A room has a link with another room if there is a corridor between them. The time to travel 1cm is 1sec. You can draw a graph to model the ADSA MAP.

Impostor can also take the vent; the map shows the link between each vent. Taking a vent do not take time.

To unmask impostors, you have the idea to compare the time to travel between any pair of room in two cases: if you are a crewmate; if you are an impostor.

Argue about the question, present the code and display the results.

1. Presents and argue about the **two models** of the map. *for the map*
2. Argue about a pathfinding algorithm to implement.
3. Implement the method and show the time to travel for any pair of rooms for both models.
4. Display the interval of time for each pair of room where the traveler is an impostor. *and compare*

Step 4: Secure the last tasks

Save the CM maps from page 13



Only few tasks remain, and you will win as a crewmate. You decide to finish the last tasks forming a pack with all remaining player. Indeed, in a pack, impostors cannot kill anyone, they will be unmasked.

The map is ADSA MAP. You need to go the quickest possible to finish all the remaining tasks before impostors distract the pack to its route. Thus, you decide to browse the map room by room, and to finish the task in the current room. A room will be visited only one time.

Argue about the question, present the code and display the results.

1. Presents and argue about the model of the map.
2. Thanks to a graph theory problem, present how to find a route passing through each room only one time.
3. Argue about an algorithm solving your problem.
4. Implement the algorithm and show a solution.

Hamilton path.

Draft the map or just rooms with tasks?