Spatial-Temporal Planning

This problem set is designed to test the skills of candidate in graph theory, planning algorithms and coding style.

Problem:

Two people (Humpty and Dumpty) are leaving for office. Their home and destinations are different but they have to pass through same tunnel to reach the destination. Only one person is allowed to pass the tunnel at once. The path followed by each person needs to be represented as graph. We will capture the travel times in the node itself; there would be one node representing the travel between home and tunnel entrance, one node to pass the tunnel and one node for travel between tunnel exit and office. So each travel graph has three nodes. A person is allowed to wait at the entrance of the tunnel if the other person is using the tunnel. To simplify, we will represent the time as integer and clock starts at 7:30am in the morning; so at time t (equals 0) they are both ready to leave. Unit of time is TU for this problem.
Humpty takes 15 TU to reach tunnel from home and 10 TU to reach office after existing tunnel
Dumpty takes 20 TU to reach tunnel from home and 5 TU to reach office after existing tunnel
It takes 5 TU to cross the tunnel for any person.

You need to plan the departure for each Humpy and Dumpty.

Note: The edges in the graph will represent the connection between nodes but has not other significant information.

Approach

- Data structures: Create node, edge and graph structure based on the information needed in your approach (30 marks). You may be
- Use algorithm of your choice (can be very simple). (20 marks)

Test

Create a rosnode to run your codes and print the whole graph on command window (travel details for each node and person).

Git: (20 points)

- Your code should be uploaded on a Git repository (private) created for this test. The name of the repo would be "FIRSTNAME_LASTNAME_FULLTIME". Name of the company "Bito" should not be there anywhere in your code or in repo.
- It is advised to do frequent commits during code development.
- Good practice: Develop each feature in a separate branch and merge back to master after module test is a bonus.
- README should be updated with procedure to run your codes.

Skills

- C++ (C++11 version)
- ROS
- Graph theory
- Motion planning
- Coding style (30 marks)

Final Submission

Submit the link of the repository when you are done.