Instructions on how to run the code is in readme.md file

The nodes at the a* algorithm:

We represented a state node as a class with 3 attributes: (current vertex, list of people, list of broken vertices).

- Current vertex = the current vertex of the state
- 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]).

Heuristic function:

We defined goal node to be a node with only zeros at the list of people.

H(current state) = take the current vertex, and all the nodes from the graph world that has people in it (we check it by the list of people), and make clique graph from those nodes. The weight on each edge should be the shortest path weight (calculated by Dijkstra) from the original path with out the broken nodes and their edges (we check it by the list of broken nodes).

After we constructed the clique graph, we run an algorithm to find his MST, and then return back the sum of all the weights at the MST.

The heuristic is admissible and we can prove it using induction:

If at the clique graph there is 1 node – that means that the heuristic function return 0, and of course its admissible.

If at the clique graph there are n nodes, lets assume that the weights of all the edges of the mst we generated from the clique graph is admissible.

If at the clique graph there are n+1 nodes, we can obtain 2 scenarios:

1. The new node reduce the mst's sum weights, and if it was admissible for n vertices, it is admissible for n+1 vertex (the more vertices there are on the graph, the farther we are from the goal).
2. The new node increased the mst's sum weights. We can observe that the edges we added from the new node to the others vertices at the clique graph, was the shortest path from the new node to each of them. So if we can't get to the goal state with "less" weight visiting the new node, and for the rest n nodes, the weight sum is admissible, the heuristic for the n+1 nodes is else admissible

Print out the results of all 3 agents with 0 , 0.000001, 0.01 T respectively as you asked us to do:

new simulation start

its AStarAgent 0 turn 0

agent 0 has moved from 0 to 1

its AStarAgent 0 turn 1

agent 0 has moved from 1 to 2

its AStarAgent 0 turn 2

agent 0 has moved from 2 to 3

its AStarAgent 0 turn 3

AStarAgent 0 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.000000

total people saved:3/3

simulation over

new simulation start

its AStarAgent 1 turn 0

agent 1 has moved from 0 to 1

its AStarAgent 1 turn 1

agent 1 has moved from 1 to 2

its AStarAgent 1 turn 2

agent 1 has moved from 2 to 3

its AStarAgent 1 turn 3

AStarAgent 1 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.000001

total people saved:3/3

simulation over

new simulation start

its AStarAgent 2 turn 0

agent 2 has moved from 0 to 1

its AStarAgent 2 turn 1

agent 2 has moved from 1 to 2

its AStarAgent 2 turn 2

agent 2 has moved from 2 to 3

its AStarAgent 2 turn 3

AStarAgent 2 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.010000

total people saved:3/3

simulation over

new simulation start

its GreedyAStarAgent 0 turn 0

agent 0 has moved from 0 to 1

its GreedyAStarAgent 0 turn 1

agent 0 has moved from 1 to 2

its GreedyAStarAgent 0 turn 2

agent 0 has moved from 2 to 3

its GreedyAStarAgent 0 turn 3

GreedyAStarAgent 0 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.000000

total people saved:3/3

simulation over

new simulation start

its GreedyAStarAgent 1 turn 0

agent 1 has moved from 0 to 1

its GreedyAStarAgent 1 turn 1

agent 1 has moved from 1 to 2

its GreedyAStarAgent 1 turn 2

agent 1 has moved from 2 to 3

its GreedyAStarAgent 1 turn 3

GreedyAStarAgent 1 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.000001

total people saved:3/3

simulation over

new simulation start

its GreedyAStarAgent 2 turn 0

agent 2 has moved from 0 to 1

its GreedyAStarAgent 2 turn 1

agent 2 has moved from 1 to 2

its GreedyAStarAgent 2 turn 2

agent 2 has moved from 2 to 3

its GreedyAStarAgent 2 turn 3

GreedyAStarAgent 2 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.010000

total people saved:3/3

simulation over

new simulation start

its RealTimeAStarAgent 0 turn 0

agent 0 has moved from 0 to 1

its RealTimeAStarAgent 0 turn 1

agent 0 has moved from 1 to 2

its RealTimeAStarAgent 0 turn 2

agent 0 has moved from 2 to 3

its RealTimeAStarAgent 0 turn 3

RealTimeAStarAgent 0 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.000000

total people saved:3/3

simulation over

new simulation start

its RealTimeAStarAgent 1 turn 0

agent 1 has moved from 0 to 1

its RealTimeAStarAgent 1 turn 1

agent 1 has moved from 1 to 2

its RealTimeAStarAgent 1 turn 2

agent 1 has moved from 2 to 3

its RealTimeAStarAgent 1 turn 3

RealTimeAStarAgent 1 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.000001

total people saved:3/3

simulation over

new simulation start

its RealTimeAStarAgent 2 turn 0

agent 2 has moved from 0 to 1

its RealTimeAStarAgent 2 turn 1

agent 2 has moved from 1 to 2

its RealTimeAStarAgent 2 turn 2

agent 2 has moved from 2 to 3

its RealTimeAStarAgent 2 turn 3

RealTimeAStarAgent 2 has been removed with a score of 2997.000000 saved 3 with the time of 3 and T was 0.010000

total people saved:3/3

simulation over