

Assignment1

Code outputs: with T=0

A* -

please enter number of agents:

1

please enter agents details:

A* V1

Agents:

```
[type:A*,
currentPosition: V1,
numOfActions: 0,
peopleEvacuated: 0,
stepsLeft = 0,
terminated =False
wating =7
strategy =[]
llimit=10000
]
```

Graph:

```
{'V1': {'p': 0, 'e': [{'v': 'V3', 'w': 2, 'blocked': False}, {'v': 'V2', 'w': 3, 'blocked': False}, {'v': 'V6', 'w': 1, 'blocked': False}, {'v': 'V7', 'w': 5, 'blocked': False}]}, 'V2': {'p': 1, 'e': [{'v': 'V1', 'w': 3, 'blocked': False}, {'v': 'V3', 'w': 4, 'blocked': False}]}, 'V3': {'p': 2, 'e': [{'v': 'V1', 'w': 2, 'blocked': False}, {'v': 'V2', 'w': 4, 'blocked': False}, {'v': 'V6', 'w': 1, 'blocked': False}, {'v': 'V4', 'w': 7, 'blocked': False}]}, 'V4': {'p': 4, 'e': [{'v': 'V3', 'w': 7, 'blocked': False}, {'v': 'V5', 'w': 4, 'blocked': False}, {'v': 'V7', 'w': 10, 'blocked': False}]}, 'V5': {'p': 5, 'e': [{'v': 'V4', 'w': 4, 'blocked': False}]}, 'V6': {'p': 0, 'e': [{'v': 'V1', 'w': 1, 'blocked': False}, {'v': 'V3', 'w': 1, 'blocked': False}]}, 'V7': {'p': 6, 'e': [{'v': 'V1', 'w': 5, 'blocked': False}, {'v': 'V4', 'w': 10, 'blocked': False}]}}
```

Astar calculation begin @@

```
Next Node: [[stateld: 0, currPos: V1, prevState: -1, prevPos: , currGval: 0, visited:[]]]
Next Node: [[stateld: 2, currPos: V3, prevState: 0, prevPos: V1, currGval: 2, visited:['V1']]]
Next Node: [[stateld: 7, currPos: V4, prevState: 2, prevPos: V3, currGval: 9, visited:['V1', 'V3']]]
Next Node: [[stateld: 1, currPos: V2, prevState: 0, prevPos: V1, currGval: 3, visited:['V1']]]
Next Node: [[stateld: 13, currPos: V3, prevState: 1, prevPos: V2, currGval: 7, visited:['V1', 'V2']]]
Next Node: [[stateld: 5, currPos: V7, prevState: 0, prevPos: V1, currGval: 5, visited:['V1']]]
Next Node: [[stateld: 22, currPos: V4, prevState: 5, prevPos: V7, currGval: 15, visited:['V1', 'V7']]]
Next Node: [[stateld: 8, currPos: V5, prevState: 2, prevPos: V3, currGval: 13, visited:['V1', 'V3']]]
Next Node: [[stateld: 21, currPos: V3, prevState: 5, prevPos: V7, currGval: 12, visited:['V1', 'V7']]]
Next Node: [[stateld: 31, currPos: V4, prevState: 21, prevPos: V3, currGval: 19, visited:['V1', 'V7', 'V3']]]
Next Node: [[stateld: 11, currPos: V5, prevState: 7, prevPos: V4, currGval: 13, visited:['V1', 'V3', 'V4']]]
Next Node: [[stateld: 32, currPos: V5, prevState: 21, prevPos: V3, currGval: 23, visited:['V1', 'V7', 'V3']]]
Next Node: [[stateld: 28, currPos: V4, prevState: 8, prevPos: V5, currGval: 17, visited:['V1', 'V3', 'V5']]]
Next Node: [[stateld: 19, currPos: V7, prevState: 13, prevPos: V3, currGval: 14, visited:['V1', 'V2', 'V3']]]
Next Node: [[stateld: 6, currPos: V2, prevState: 2, prevPos: V3, currGval: 6, visited:['V1', 'V3']]]
Next Node: [[stateld: 20, currPos: V2, prevState: 5, prevPos: V7, currGval: 13, visited:['V1', 'V7']]]
Next Node: [[stateld: 41, currPos: V4, prevState: 19, prevPos: V7, currGval: 24, visited:['V1', 'V2', 'V3', 'V7']]]
Next Node: [[stateld: 45, currPos: V7, prevState: 6, prevPos: V2, currGval: 14, visited:['V1', 'V3', 'V2']]]
Next Node: [[stateld: 49, currPos: V5, prevState: 41, prevPos: V4, currGval: 28, visited:['V1', 'V2', 'V3', 'V7', 'V4']]]
```

Astar calculation finished @@

New Strategy: abs - ['V1', 'V2', 'V3', 'V7', 'V4', 'V5'], full - ['V1', 'V2', 'V3', 'V1', 'V7', 'V4', 'V5']

Next step: V2

Next step: V3

Next step: V1

Next step: V7

Next step: V4

Next step: V5

Well Done!!! Agent: A*

evacuated 18 people! And it took 35.0 rounds.

RTA (L=5) –

please enter number of agents:

1

please enter agents details:

rta V1

Agents:

```
[type:rta,  
currentPosition: V1,  
numOfActions: 0,  
peopleEvacuated: 0,  
stepsLeft = 0,  
terminated =False  
wating =7  
strategy =[]  
llimit=5  
]
```

Graph:

```
{'V1': {'p': 0, 'e': [{'v': 'V3', 'w': 2, 'blocked': False}, {'v': 'V2', 'w': 3, 'blocked': False}, {'v': 'V6', 'w': 1, 'blocked': False}, {'v': 'V7',  
'w': 5, 'blocked': False}]}, 'V2': {'p': 1, 'e': [{'v': 'V1', 'w': 3, 'blocked': False}, {'v': 'V3', 'w': 4, 'blocked': False}]}, 'V3': {'p': 2, 'e': [{'v':  
'V1', 'w': 2, 'blocked': False}, {'v': 'V2', 'w': 4, 'blocked': False}, {'v': 'V6', 'w': 1, 'blocked': False}, {'v': 'V4', 'w': 7, 'blocked': False}]},  
'V4': {'p': 4, 'e': [{'v': 'V3', 'w': 7, 'blocked': False}, {'v': 'V5', 'w': 4, 'blocked': False}, {'v': 'V7', 'w': 10, 'blocked': False}]}, 'V5': {'p':  
5, 'e': [{'v': 'V4', 'w': 4, 'blocked': False}]}, 'V6': {'p': 0, 'e': [{'v': 'V1', 'w': 1, 'blocked': False}, {'v': 'V3', 'w': 1, 'blocked': False}]},  
'V7': {'p': 6, 'e': [{'v': 'V1', 'w': 5, 'blocked': False}, {'v': 'V4', 'w': 10, 'blocked': False}]}}
```

Astar calculation begin @@

Next Node: [[stateld: 0, currPos: V1, prevState: -1, prevPos: , currGval: 0, visited:[]]]

Next Node: [[stateld: 2, currPos: V3, prevState: 0, prevPos: V1, currGval: 2, visited:['V1']]]

Next Node: [[stateld: 7, currPos: V4, prevState: 2, prevPos: V3, currGval: 9, visited:['V1', 'V3']]]

Next Node: [[stateld: 1, currPos: V2, prevState: 0, prevPos: V1, currGval: 3, visited:['V1']]]

Next Node: [[stateld: 13, currPos: V3, prevState: 1, prevPos: V2, currGval: 7, visited:['V1', 'V2']]]

Astar calculation finished @@

New Strategy: abs - ['V1', 'V7'], full - ['V1', 'V7']

Next step: V7

Astar calculation begin @@

Next Node: [[stateld: 19, currPos: V7, prevState: -1, prevPos: , currGval: 0, visited:[]]]

Next Node: [[stateld: 21, currPos: V3, prevState: 19, prevPos: V7, currGval: 7, visited:['V7']]]

Next Node: [[stateld: 25, currPos: V4, prevState: 21, prevPos: V3, currGval: 14, visited:['V7', 'V3']]]

Next Node: [[stateld: 26, currPos: V5, prevState: 21, prevPos: V3, currGval: 18, visited:['V7', 'V3']]]

Next Node: [[stateld: 20, currPos: V2, prevState: 19, prevPos: V7, currGval: 8, visited:['V7']]]

Astar calculation finished @@

New Strategy: abs - ['V7', 'V2', 'V3'], full - ['V7', 'V1', 'V2', 'V3']

Next step: V1

Next step: V2

Next step: V3

Astar calculation begin @@

Next Node: [[stateld: 33, currPos: V3, prevState: -1, prevPos: , currGval: 0, visited:[]]]

Next Node: [[stateld: 34, currPos: V4, prevState: 33, prevPos: V3, currGval: 7, visited:['V3']]]

Next Node: [[stateld: 36, currPos: V5, prevState: 34, prevPos: V4, currGval: 11, visited:['V3', 'V4']]]

Astar calculation finished @@

New Strategy: abs - ['V3', 'V4', 'V5'], full - ['V3', 'V4', 'V5']

Next step: V4

Next step: V5

Well Done!!! Agent: rta

evacuated 18 people! And it took 35.0 rounds.

greedy –

please enter number of agents:

1

please enter agents details:

greedy V1

Agents:

```
[type:greedy,
currentPosition: V1,
numOfActions: 0,
peopleEvacuated: 0,
stepsLeft = 0,
terminated =False
wating =7
strategy =[]
llimit=1
]
```

Graph:

```
{'V1': {'p': 0, 'e': [{'v': 'V3', 'w': 2, 'blocked': False}, {'v': 'V2', 'w': 3, 'blocked': False}, {'v': 'V6', 'w': 1, 'blocked': False}, {'v': 'V7', 'w': 5, 'blocked': False}]}, 'V2': {'p': 1, 'e': [{'v': 'V1', 'w': 3, 'blocked': False}, {'v': 'V3', 'w': 4, 'blocked': False}]}, 'V3': {'p': 2, 'e': [{'v': 'V1', 'w': 2, 'blocked': False}, {'v': 'V2', 'w': 4, 'blocked': False}, {'v': 'V6', 'w': 1, 'blocked': False}, {'v': 'V4', 'w': 7, 'blocked': False}]}, 'V4': {'p': 4, 'e': [{'v': 'V3', 'w': 7, 'blocked': False}, {'v': 'V5', 'w': 4, 'blocked': False}, {'v': 'V7', 'w': 10, 'blocked': False}]}, 'V5': {'p': 5, 'e': [{'v': 'V4', 'w': 4, 'blocked': False}]}, 'V6': {'p': 0, 'e': [{'v': 'V1', 'w': 1, 'blocked': False}, {'v': 'V3', 'w': 1, 'blocked': False}]}, 'V7': {'p': 6, 'e': [{'v': 'V1', 'w': 5, 'blocked': False}, {'v': 'V4', 'w': 10, 'blocked': False}]}}
```

Astar calculation begin @@

Next Node: [[stateld: 0, currPos: V1, prevState: -1, prevPos: , currGval: 0, visited:[]]]

Astar calculation finished @@

New Strategy: abs - ['V1', 'V5'], full - ['V1', 'V3', 'V4', 'V5']

Next step: V3

Next step: V4

Next step: V5

Astar calculation begin @@

Next Node: [[stateld: 5, currPos: V5, prevState: -1, prevPos: , currGval: 0, visited:[]]]

Astar calculation finished @@

New Strategy: abs - ['V5', 'V2'], full - ['V5', 'V4', 'V3', 'V2']

Next step: V4

Next step: V3

Next step: V2

Astar calculation begin @@

Next Node: [[stateld: 7, currPos: V2, prevState: -1, prevPos: , currGval: 0, visited:[]]]

Astar calculation finished @@

New Strategy: abs - ['V2', 'V7'], full - ['V2', 'V1', 'V7']

Next step: V1

Next step: V7

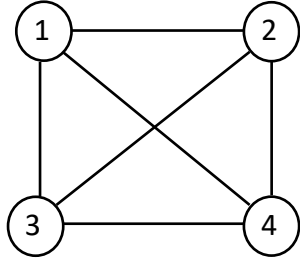
Well Done!!! Agent: greedy

evacuated 18 people! And it took 45.0 rounds.

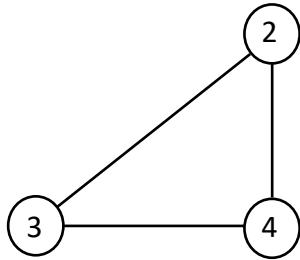
Heuristic function:

For a state, containing (currentVertex, visitedVertices) the function is calculating the MST for the induced graph of the nodes left to visit (the vertices with people) and the current vertex.

For example –



When currentVertex = 2 and visitedVertices = [1], we will calculate the MST of



and the sum of the MST edges weight will be the heuristic function value of (2,[1]) state.

The rationale of choosing this function is we are taking the vertices we have to go through, and we know that in the best-case scenario the shortest path through them all is the weight of the MST, and in other cases it will be larger, because we have no guaranty that the MST is arranged in a way we can travel him by traveling 1 time on each edge.

Also, we guaranty that $h(\text{Goal}) = 0$, because the state will mean that we visited in all needed vertices, and the induced graph will be of the current vertex alone, so the MST will be with a weight of 0.

Different T:

T	A*	RTA (L=5)	Greedy
0	35	35	45
0.000001	36	38	48
0.01	36	38	48

(time unit)