AIFA ASSIGNMENT 1

MULTI AGENT PATHFINDING

COMPILED BY:

Ashish Kumar	19IM10008
Harsh Agarwal	19MF10016
Arka Mitra	17EC10067

Consider a warehouse management system by a group of robots. A group of robots need to pick up items from designated places, deliver those in desired locations, and finally the robots go to their respective destination location. This can be seen as a multiagent system with shared tasks as objective. Assume that the entire planning needs to be done in a 2D grid representation of size n × m. Each location (Li) on the warehouse floor can be denoted by a pair of integers Li=(xi, yi). Each cell on the grid can be categorized in one of the following ways (see diagram below) - source location (P1, P2 ...), destination location (D1,D2,...), temporary storage location (TS1,TS2,....), obstacle (black square),normal (rest of the cells). Source & destination denote pick-up and drop locations respectively.

Temporary storage location denotes the place where robots can keep some items. Obstacles represent a location where no robot can move. Rest of the cells are considered normal cells.

Let there be k number of robots and r number of tasks. The details of robot location and tasks are provided as per the following table.

Let's take k=4 robots and r= 4 tasks for example. Have a look at it.

ROBOTS	LOCATION	
	Initial	Final
ROBOT1	R1	E1
ROBOT2	R2	E2
ROBOT3	R3	E2

ROBOT4	R4	E3
TASKS	LOCATION	
	Pickup	Deliver
Task1	P1	D1
Task2	P2	D3
Task3	P3	D3
Task4	P4	D2

Assume that a robot can move at most one cell (either vertically or horizontally) at a time step, a normal cell can be occupied by at most one robot. Source, destination, temporary storage locations can accommodate multiple robots simultaneously. Our target here is to develop a work schedule that minimizes the time to complete all tasks. You need to develop both optimal as well as heuristic algorithms.

Let's take a grid of size(m*n) and assign start locations and end locations of robot and assign pickup and delivery locations of a task according to problem. Assign storage stations.

Take any m and n values. I took 6*10 grid but you can take whatever you need.

R1			P2		TS2				
						E1			D3
		D3		P1				E3	
	R2		TS1						P4
	D2							D1	
R4		R3			E2	TS3		P3	

Here black coloured boxes are blocked cells where no robot can go through that cell. TS texted are storage stations where robots can store. Blue coloured are starts and end locations of robots and Red coloured are pick-up and end locations of tasks.

The following assumptions are considered while solving the path finding problem. i. Each Agent can move a single step at a single unit of time.

ii. the static obstacles will not be moving.

Iii. Dynamic obstacles need to be handled by the agents.

Ii. The robots may use the temporary location not in all cases and will have prior knowledge about its usage.

Explanation of our Assignment:

In our source code, we have two input files that are input.yaml file and input1.yaml file and a output file named output.yaml.

The code needs to be run like python3 input.yaml output.yaml

INPUT file:

It is the input file which we have to predefine the No.of Robots which are available to complete the all tasks.We have to predefine the Robot's start and end locations and Task's pickup and delivery locations.

For any agent let's say agent0, We have 4 tasks as following In this our format in INPUT1 file:

```
goal: [7, 2]
name: agent0
start: [2, 5]
```

It means agent0 starts from the start position and reaches to this goal position. Like this,we have the following four tasks as mentioned below.

Task1: robot start position to pick-up location. Task2: pickup location to storage station.

Task3: storage station to delivery location

Task4:delivery location to final location of robot.

In this file,we have other inputs which are obstacle cells where robots can not progress through that cell.

In our input file we just take two robots but we can take any number of robots we want. We can make our desired grid size and can make any number of obstacles.Care must be taken in assigning start location,end location,pickup location,end location as these can't be obstacle cells. This is our input file.

OUTPUT FILE:

In the output file, Every agent starts from their starting position at time t=0 secs to do the Task1 as we mentioned in the input file. Output is in the format that at time t,x and y coordinates of a robot are printed as below. After completion of Task1 by all robots, Task2 will be initiated at time t=x where x is a time taken by the corresponding agent to complete its previous task i.e, Task1 here. Like this, by completion of all tasks by all robots, we will get time taken by a robot to complete all of his tasks.

Agent0: Agent1:

```
- t: 0
  x: 2
  y: 5
- t: 1
  x: 3
  y: 5
```

```
- t: 0
  x: 3
  y: 1
-t: 1
  x: 3
  y: 2
```

This means that at time t=0s,agent0 is at(2,5) and agent1 is at(3,1). At time t=1s,agent0 is at(3,5) and agent1 is at(3,2).