CSC520 Artificial Intelligence Assignment 1

Sai Vishnu Muvvala
ID: 200258411
Email: smuvval@ncsu.edu

Question 1:
Scenario 1: Intelligent Monitor
Part 1.
Performance: Make the video recording only when targets are moving. So a good agent will,

1. Capture at the maximum extent possible(no standby mode), when targets move.
2. Have less false positives- Have no cases where video is recorded even when there's no moving target.

Environment: Entirety of storehouse covering every corner and people in it.
Actuators: Video recorder that can be commanded to record/standby automatically(no manual operations here - soft-switch).
Sensors: Motion tracking sensors must be installed in all parts of storehouse.
Features of environment:
Fully observable: Unless there is some stationary obstruction to sensors, the motion sensors must be able to detect any moving objects in the storehouse, thus making it fully observable.

Multi agent: The agents actions are completly dependent on moving objects such as humans. Hence it's multi agent environment

Deterministic: Although there's uncertainty pertaining to randomness in moving activity, as this is a fully observable multiagent environment, the uncertainty can be ignored.

Episodic: As the next move record/standby doesn't depend on the current move, the environment is Episodic.

Dynamic: The environment containing people, moving objects might keep moving all through, making it a dynamic environment.

Discrete: The environment can be divided into 2 states. With motion and without motion. There's no state beyond this. Hence, it's discrete.

Known: As everything about storehouse is known and there's nothing changing about it, it's known.
Part 2. Agent type:
Simple Reflex Agent can solve this problem. Simple reflex agents are about storing list of to-do actions, and perform them depending on the situation in the environment. For every predicted state of the environment, this agent has got a response action.

In the current problem, there are two states of environment.

1. No moving targets are present in the environment.
2. Moving targets are present.

For 1, the actuators standby. For 2, actuators record. Hence, a simple reflex agent is enough for this problem.

Scenario 2: Deep Sea Drone

Part 1.

Performance: As it's an agent that does many things, there are various parts of the agent that must be considered for evaluating performance.

1. Proximity of agent to planned destination: Agent must go to the right depth distance which was pre-specified.
2. Salinity check - Agent must measure salinity correctly.
3. Temperature check - Agent must measure temperature correctly.
4. Data transmission after returning to surface - Agent must rightly and safely return to the surface and transmit the identified recorded values of data.

Environment: Ocean, which might be having currents.

Actuators: Deep sea navigation motor system that can dive to desired depth and return back safely. Data transmitter for transmitting data after returning back.

Sensors: Conductivity meter to measure salinity, Thermometer to measure temperature, altitude sensor.

Features of environment:

Partially observable: The environment of deep sea drone is ocean. The entire of ocean cannot be seen clearly. Only some neighbouring distance can be observed. Hence, partially observable.

Multi agent: The diving motion of the agent depends on neighbouring objects like rocks, fishes and currents too. Hence, multiagent.

Stochastic: As there's some randomness of fishes, currents in the environment, it is stochastic.

Sequential: As current motion decides next motion in traveling to the desired depth, it's sequential.

Dynamic: The ocean environment is dynamic, changing every moment, because of the life inside it. Hence, dynamic.

Continuous: Over time, the speed and location of the drone and other life around it sweeps through a range of continuous values. Hence, it's continuous.

Known: Even though the ocean space is not heavily explored, we can make some assumptions about the environment in it. Hence, Known.

Part 2: Agent type:

A utility based agent is required here. The aim of the drone is to dive, record, comeback, transmit data. For doing all these actions, the drone has to safely return. The drone has to bear the currents of the ocean too. So, at every point, the drone has to look out for the actions that can keep it safe(happy) and work well. Hence, a utility based agent is required.

Scenario 3: Tax Assistance

Part 1.

Performance:

1. Error-free: The agent must recommend correct forms but not the wrong ones.
2. Cover every case in the taxing system.

Environment: The environment contains

1. Tax laws
2. Deadlines
3. Users, with their information

Actuators:

1. Display a list of forms and guidelines for users to follow and fill.
2. Generate forms in the UI for users to fill.

Sensors: Inputs from keyboard/mouse (or) mic with natural language processor.

Features of environment:

Fully observable: All the information is gathered from the user. Hence fully observable.

Single agent: The tax requirements of one individual doesn't depend on another. Hence it's single agent.

Deterministic: There's no randomness to this environment. Also, requesting data and asking to fill forms is a step-by-step process. As next state is determined by current state, it's deterministic.

Sequential: Here, we know that a user responds to questions asked by agent. One question generally depends on another question. Like, if user responds to the question "Did you file state tax" and user responds "No", agent shall ask more questions about things pertaining to state tax. As each action defines next action, it is sequential.

Static: When the agent is processing on the information it received, the environment doesn't suddenly change. Hence, it's a static environment.

Discrete: In this environment, time is handled discretely. Agent asks user some questions and user responds. All of this is discrete.

Known: As all the laws of the tax system and the user information is acquired, everything is known here.

Part 2: Agent type:

Model-based Reflex Agent is required here. According to the problem specification, the agent asks a series of questions to the users about their current financial situation. So, memory is required for the agent. With the obtained knowledge, the agent predicts the situation of the user. Using it's knowledge of Tax laws, agent can work on a series of if-else blocks to match the case of the current user, to recommend corresponding forms.

Scenario 4: **Intelligent Infrastructure Management**
Part 1.
Performance: Manage the resources effectively to:
1. Allot resources when required by users.
2. Reduce power usage.
3. Reduce response time.

Environment: Environment contains:
1. Online resources, such as virtual machines, soft-switch hardware resources such as networking kits.
2. Users/system who/which sought for obtaining/relinquish online resources.
3. Input service requests/renounces for online resources.

Actuators:
1. Hardware cables connected to servers where online resources are hosted. Commands to turn on/off are sent from agent to these resources using the hardware.
2. Soft switch commands(using telnet/UDP) sent to virtual resources that can be controlled online.

Sensors: Real-time data pertaining to current status of online resources, current users, queued number of requests, current power consumption.
Features of environment:
Fully observable: The environment is fully observable as the agent is able to view the status of the entire resources, power consumption, users.

Multi agent: The users request for online resources. Even though the agent has the best model to save electricity and reduce wait time, a huge number of users might unpredictably request for resources. As the performance of the agent depends on users, it's multi agent.

Stochastic: Even though there's prediction model for input service request rate, there's some internal randomness present to service requests for online resources. The requests might go up any moment. Hence, it's stochastic.

Sequential: If there are no users to systems, the agent might turn off some of the online resources. Turning off resources shall effect next states and might increase wait time, if there's sudden increase in demand. Each action has effect on next one. Hence, it's sequential.

Dynamic: Users might rapidly increase/decrease. As the current state of usage of online resources keeps changing, the environment is dynamic.

Continuous: As there are huge amounts of online resources and huge number of users, users might be constantly acquiring/renouncing resources, over time, the free/acquired resources sweep through a range of continuous values, it is continuous.

Unknown: If there's a working stochastic prediction of service input request rate (for online resources), the environment can be known. If the agent hasn't learned the way input request rate might go up or down, it will be an unknown environment. If this intelligent agent is used in one company, where agent has been used for so long and agent can predict the service rate at different situations, it will be known environment for it. On the other hand, if the same agent is installed in a new organization, whose service requests are unpredictable, it is unknown environment.

Part 2: Agent type:

A utility based agent is required here. For Intelligent Infrastructure Management, there's no specific goal, like, save 10MW electricity or reduce wait time to 5sec. The agent aims to do it's best to balance between saving electricity and also reduce waiting time. There is no specific goal here, but the target of evaluating the 'happy state'. Hence, utility based agent is required.

Scenario 5: Robot Soccer Player

Part 5.

Performance: As the agent is sophisticated and complex, performance is a measure of various factors such as:

1. Make as many goals as possible.
2. Coordinate with other robots of same team.
3. Ball tracker - Ability of robots to rightly and swiftly predict the direction/trajectory of the ball.
4. Ability of robots to rightly distinguish opponents from same team.
5. Ability of robots to predict the location of unknown robots(which are not in sensor zone), by recognizing their last found location.

Environment: Environment of each robot contains:

1. Other robots.
2. Football and goal location.
3. Winds(if any) - If winds are present, they play a major role in deflecting the trajectory of a ball.

Actuators: Robotic legs that can move, kick, and toss a football.

Sensors:

1. Sensors to identify football.
2. Sensors to identify goal location.
3. Proximity sensors to identify nearby objects, robots.

Features of environment:

Partially observable: As a robot can only sense other robots within it's range but not the entire of football field, the environment for agent is partially observable.

Multi agent: As robots are both competing and supporting, it's a multiagent environment.

Stochastic: With the direction of wind flow, the movement of football might suddenly change. Also, there's some randomness in the sense of other players who are not currently in the observable zone. Hence it's stochastic environment.

Sequential: Every action performed by robot, such as move forward, move backward, shall change the state of its own and of the environment too. Hence, it's sequential.

Dynamic: As a robot assesses current state of environment, the environment around it keeps changing. The other players might be moving, ball might be moving. Hence it's dynamic.

continuous: The movement of the football and other robots must be tracked every moment. Hence it's continuous.

Known: Even though robot doesn't exactly know the complete state of environment, it knows how the environment works. Hence it's known.

Part 2: Agent type:

A **goal based agent** is required here. The sole aim of the robot soccer player is to make as many goals as possible. For this, it has to coordinate with other robots of same team and make goals. If the problem statement was to have better score than other team, a utility based agent might be required. This is because, to have better score, a player should not just target to hit the goals. He must also play defence against opposition. The aim of a player is to both hit the goals and defend the opposition. Based on the current location of the football and other players, a robot soccer player must evaluate who has the chance of attaining the goal. If it's his team, he has to support it. If opposition has more chance, he has to defend. The target essentially is to attain the happiest state from the current state.

Question 2:

Given that for each action, there is a slip chance for the agent to move in the opposite direction. Given sequence of actions are : {right,right,top,left,left}

Initial position- {a}
After 1st action(right)- {a,b}
After 2nd action(right)- {a,b,c}
After 3rd action(top)- {a,b,c,f}
After 4th action(left)- {a,b,c,f,g,e}
After 5th action(left)- {a,b,c,f,g,e,h}

Question 3:

Part a: The best move for max is to opt for the move that gives him maximum value(8).

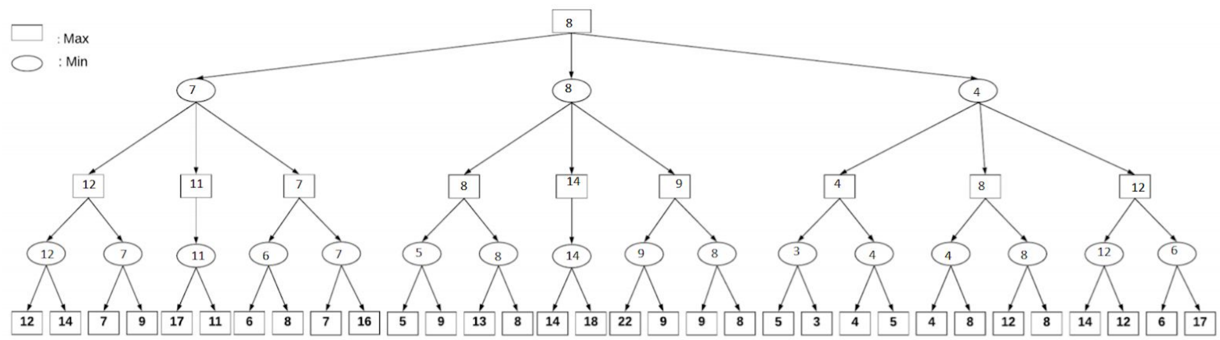

Figure 1: Game Tree for Q3.a

Part b:

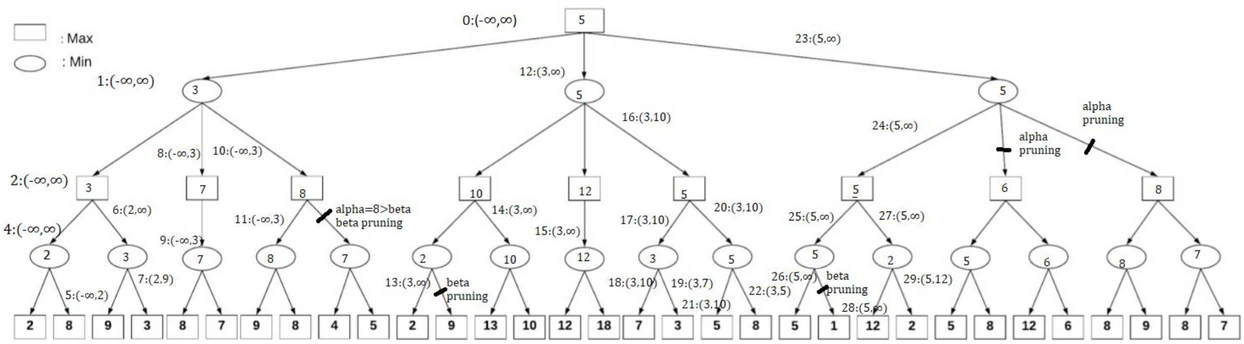

Figure 2: Game Tree for Q3.b

Part c: The best move for player A is to opt for the step that gives him maximum value=5.75.

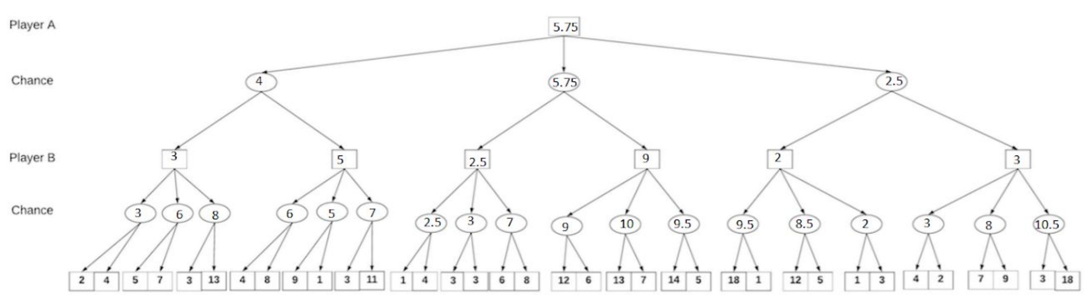

Figure 3: Game Tree for Q3.c

Given that there is equal probability for every possible outcome, down the line. As the process is stochastic and there is ½ chance of every outcome, the nodes in 'chance' layer are filled with weighted values, where every outcome is given equal weight.

Value in chance node=Σ(1/number of children)*(child value)

The available options for player A are with values 4, 5.75, 2.5. To maximize profit, player A has to pick the move that gives him highest value of 5.75.

Question 4:

4.1 Heuristic solution:

As this is a number maze problem, heuristics are evaluated for vertices of the maze. For a vertex, heuristic is defined as:

heuristic(vertex)

1. If the vertex is the destination vertex, then heuristic=0.
2. If the vertex is within reachable "**range**" of destination vertex, then heuristic=1.
3. If the vertex is not in the reachable range of destination vertex, then heuristic=2.

What is range here? If we can move from a vertex v1 to vertex v2, then v2 is in the range of v1.

Explanation:

In a number maze, we know that, **potentially**, we every vertex is reachable from every other vertex in a maximum number of 2 moves. If value inside a vertex is big enough to reach destination from source, we can move in just one move(with some relaxed rules). Hence, if the destination is reachable within one move, heuristic is 1. Else, heuristic is 2.

Approach to creating heuristic:

Relaxation method is used in creating this heuristic. We know that the rigid rule present in the number maze problem is

"*In a single move, we can make horizontal or vertical steps from one cell to another*"

The rule is relaxed as:

"In a single move, we can make both horizontal and vertical steps"

Thus, the rule pertaining to rigid movement is relaxed.

We know that, relaxed state space is superset of problem state space. Hence, solution for problem state space is also a part of this superset and our heuristic can be applied.

Example move: In below 4x4 maze, from cell (3,3), we can make 2 moves. These moves can be horizontal, vertical or both.

	1	2	3	4
1	2	1	1	3
2	2	1	2	3
3	1	1	2	3
4	3	1	3	1

Hence, next cell could be one of:
Horizontal: (3,1)
Vertical: (1,3)
Both horizontal and vertical: (2,2), (2,4), (4,2), (4,4)

Why is this heuristic admissible?
Admissible infers to optimization. The heuristic must never overestimate. It must always be optimistic. We know that, *potentially* we can move from any cell to any other cell in a maximum of 2 moves only (if values are large enough).
Above stated heuristic shall return a maximum value of 2, when one cell is not in reachable range of another. Hence, the heuristic is not overestimating. In this sense, it is admissible.

Limitations of this heuristic:
The heuristic doesn't guarantee consistency. Say there are two cells c1, c2 that are not in range of destination D. When evaluating heuristic, both cells c1, c2 shall have same heuristic value 2. It might be that cell c1 is in range of another cell c3, which is in range of destination. So, from c1, we can reach destination in 2 steps (c1->c3->destination). Whereas cell c2 is far away from destination, where it takes more than 2 moves to reach destination. But our heuristic gives assigns same weight ('2') for both c1 and c2. Hence, the heuristic is not essentially consistent.

Question 4.2: Code files added separately. Please check myfile.java
In order to run the program, please check readme.txt to identify the commands to run.

Question 4.3:
4x4 Maze:
DFS:

a. ***The shortest path(First path identified from source to destination, as clarified by professor)***

For 4x4 maze, the first path identified by DFS is: Cost: 11 cells=10 moves
0,0
2,0
1,0
3,0

3,3
3,2
0,2
0,1
1,1
2,1
3,1

b. Number of states expanded:
For DFS, number of states expanded=15

BFS:

a. First path identified by BFS: Cost=4 cells/3 moves

0,0
2,0
2,1
3,1
b. Number of states expanded=12
Best First:

a. First path identified by Best First: Cost =4 cells/3 moves

0,0
2,0
2,1
3,1

b. Number of states expanded=5

A star:

a. First path identified by A star: Cost=4 cells/3 moves

0,0
2,0
2,1
3,1

b. Number of states expanded=5

Unique paths:
Best Route for given commands:
0,0
2,0
2,1
3,1
Cost: 4

Number of unique paths: 9
Number of states expanded: 415

6x6 Maze:
DFS:

a. First path identified by DFS: Cost= 21 cells/ 20moves

0,0	2,5	5,4
5,0	0,5	2,4
3,0	4,5	2,0
3,3	4,1	1,0
0,3	1,1	1,3
2,3	1,4	5,3
2,1	3,4	5,5

b. Number of states expanded=31

BFS:

a. First path identified by BFS: Cost= 14 cells/13 moves

0,0	1,4	1,0
0,5	3,4	1,3
4,5	5,4	5,3
4,1	2,4	5,5
1,1	2,0	

b. Number of states expanded=35

Best First:

a. First path identified by Best First: Cost= 18 cells/17 moves

0,0	4,5	2,4
5,0	4,1	2,0
3,0	1,1	1,0
3,3	1,4	1,3
0,3	3,4	5,3
0,5	5,4	5,5

b. Number of states expanded=55

A star:

a. First path identified by A star: Cost= 14 cells/13 moves

0,0	1,4	1,0
0,5	3,4	1,3
4,5	5,4	5,3
4,1	2,4	5,5
1,1	2,0	

b. Number of states expanded=108

Unique Paths:
Best Route for given commands:

0,0	1,4	1,0
0,5	3,4	1,3
4,5	5,4	5,3
4,1	2,4	5,5
1,1	2,0	

Cost: 14
Number of unique paths: 79
Number of states expanded: 8403

8x8 Maze:
DFS:

a. First path identified by DFS: Cost=23 cells/22 moves

0,0	5,2	1,2
7,0	5,5	1,0
7,7	3,5	2,0
0,7	3,3	2,4
1,7	4,3	6,4
2,7	6,3	6,5
2,2	5,3	4,5
0,2	1,3	

b. Number of states expanded=49

BFS:

a. First path identified by BFS: Cost=8 cells/7 moves

0,0
0,7
0,6
0,3

6,3
6,4
6,5
4,5

b. Number of states expanded=40

Best First:

a. First path identified by Best First: Cost=14 cells/13 moves

0,0
7,0
7,7
0,7
1,7
2,7
2,2
0,2
5,2
5,5
3,5
3,3
4,3
4,5

b. Number of states expanded=31

A star:

a. First path identified by A star: Cost=9 cells/8 moves

0,0
0,7
0,6
0,3
6,3
7,3
3,3
4,3
4,5

b. Number of states expanded=61

Unique Paths:

Best Route for given commands:
0,0
0,7
0,6
0,3
6,3
6,4
6,5
4,5
Cost: 8

Number of unique paths: 1110
Number of states expanded: 108513

12x12a Maze:
DFS:

a. First path identified by DFS: Cost=41 cells/40 moves

0,0
2,0
4,0
3,0
1,0
7,0
6,0
6,6
1,6
7,6
2,6
3,6
4,6
0,6
0,5
4,5
10,5
10,11
5,11
2,11
2,8
8,8
3,8
4,8
10,8
9,8
7,8
5,8
5,5
1,5
1,2
4,2
0,2
6,2
8,2
5,2
2,2
7,2
7,1
2,1
5,1

b. Number of states expanded=41

BFS:

a. First path identified by BFS: Cost=7 cells/6 moves

0,0
2,0
4,0
4,1
7,1
2,1
5,1

b. Number of states expanded=98

Best First:

a. First path identified by Best First: Cost=7 cells/6 moves

0,0
2,0
4,0
4,1
7,1

2,1
5,1

b. Number of states expanded=9

A star:

a. First path identified by A star: Cost=7 cells/6 moves

0,0
2,0
2,2
7,2
7,1
2,1
5,1

b. Number of states expanded=23

12x12b Maze:
DFS:

a. First path identified by DFS: Cost=14 cells/13 moves

0,0
6,0
4,0
11,0
3,0
8,0
8,8
0,8
3,8
11,8
10,8
10,4
2,4
5,4

b. Number of states expanded=18

BFS:

a. First path identified by BFS: Cost=8 cells/7 moves

0,0
6,0
8,0
8,8

0,8
0,11
5,11
5,4

b. Number of states expanded=87

Best First:

a. First path identified by Best First: Cost=14 cells/13 moves

0,0
6,0
4,0
4,7
10,7
10,1
4,1
3,1
3,8
11,8
10,8
10,4
2,4
5,4

b. Number of states expanded=24

A star:

a. First path identified by A star: Cost=8 cells/7 moves

0,0
6,0
6,2
10,2
10,5
10,4
2,4
5,4

b. Number of states expanded=48

25x25 Maze:
DFS:

a. First path identified by DFS: Cost=347 cells/346 moves

Note: The path below is in 3-column structure. Per page, column1 is followed by column2. Column 2 is followed by column3. After 3 columns are over in a page, cells continue in next page.

0,0
9,0
1,0
2,0
6,0
12,0
10,0
14,0
16,0
22,0
11,0
4,0
4,2
13,2
9,2
7,2
17,2
6,2
2,2
8,2
8,1
18,1
12,1
4,1
2,1
7,1
17,1
22,1
22,11
11,11
9,11
8,11
7,11
17,11
17,21
9,21
5,21
11,21
4,21
1,21
1,18
11,18
4,18
7,18
10,18
5,18
2,18
0,18
9,18
17,18
6,18
6,13
3,13
10,13
9,13
0,13
0,4
6,4
5,4
16,4
10,4
2,4
12,4
22,4
22,10
13,10
4,10
10,10
1,10
0,10
0,6
3,6
8,6
7,6
6,6
15,6
11,6
1,6
1,16
7,16
3,16
11,16
6,16
15,16
5,16
9,16
13,16
21,16
12,16
12,7
10,7
0,7
8,7
16,7
13,7
7,7
2,7
5,7
5,2
3,2
3,5
12,5
9,5
6,5
1,5
10,5
14,5
11,5
21,5
20,5
20,15
18,15
12,15
10,15
1,15
7,15
7,4
18,4
24,4
24,2

23,2
22,2
19,2
11,2
11,10
5,10
2,10
2,8
8,8
5,8
5,5
5,9
3,9
7,9
6,9
1,9
9,9
8,9
13,9
2,9
2,20
0,20
11,20
4,20
10,20
9,20
1,20
7,20
5,20
12,20
22,20
19,20
15,20
8,20
18,20
16,20
13,20
13,18
16,18
16,8
10,8
1,8
3,8
3,3
1,3
1,1
11,1
21,1
10,1
15,1
19,1
19,8
11,8
12,8
13,8
14,8
6,8
6,14
13,14
9,14
0,14
10,14
21,14
12,14
3,14
3,11
12,11
6,11
15,11
23,11
19,11
21,11
14,11
16,11
16,1
16,10
14,10
17,10
23,10
15,10
15,0
18,0
8,0
19,0
19,3
10,3
2,3
11,3
4,3
4,6
14,6
12,6
9,6
18,6
18,17
11,17
8,17
5,17
12,17
16,17
16,6
24,6
23,6
20,6
17,6
17,3
14,3
24,3
18,3
23,3
23,8
21,8
21,19
10,19
6,19
16,19
14,19
11,19
7,19
5,19
13,19
24,19
24,8
22,8
22,14
11,14
7,14
5,14
4,14

4,7
4,4
3,4
3,0
7,0
17,0
21,0
21,4
21,9
20,9
12,9
11,9
0,9
0,5
0,16
2,16
8,16
8,23
5,23
16,23
11,23
20,23
10,23
14,23
12,23
15,23
7,23
17,23
17,15
17,5
8,5
8,3
8,12
3,12
12,12
21,12
17,12
7,12
6,12
5,12
13,12
23,12
14,12
16,12
9,12
20,12
20,18
14,18
15,18
15,13
20,13
11,13
5,13
4,13
14,13
14,24
9,24
6,24
12,24
1,24
2,24
13,24
13,17
22,17
14,17
23,17
19,17
24,17
24,15
16,15
16,14
8,14
14,14
15,14
20,14
19,14
19,12
10,12
10,21
8,21
0,21
0,15
0,22
10,22
9,22
2,22
12,22
14,22
19,22
19,16
19,6
10,6
21,6
21,15
21,20
24,20
24,16
24,7

b. Number of states expanded=725

BFS:

a. First path identified by BFS: Cost=9 cells/8 moves

0,0
9,0
17,0
13,0
19,0

16,0
16,6
24,6
24,7

b. Number of states expanded=509

Best First:

a. First path identified by Best First: Cost=172 cells/171 moves

0,0
9,0
1,0
2,0
6,0
12,0
10,0
14,0
16,0
22,0
11,0
18,0
18,10
14,10
14,7
3,7
12,7
10,7
20,7
9,7
1,7
8,7
16,7
13,7
7,7
7,12
6,12
5,12
13,12
23,12
14,12
12,12
21,12
17,12
17,22
16,22
15,22
8,22
0,22
10,22
9,22
2,22
12,22
14,22
19,22
13,22
18,22
18,14
18,7
18,15
12,15
10,15
1,15
7,15
7,4
18,4
12,4
22,4
16,4
10,4
10,12
19,12
19,21
19,11
17,11
17,21
17,13
9,13
18,13
14,13
3,13
10,13
11,13
5,13
6,13
6,16
15,16
15,6
11,6
21,6
12,6
9,6
18,6
7,6
6,6
6,15
17,15
17,5
8,5
6,5
1,5
10,5
14,5
11,5
21,5
20,5
20,15
22,15
19,15

11,15
11,4
9,4
9,5
12,5
15,5
23,5
18,5
16,5
16,12
16,19
14,19
11,19
7,19
5,19
13,19
2,19
10,19
10,23
6,23
16,23
11,23
2,23
2,17
5,17
12,17
8,17
11,17
14,17
23,17
19,17
24,17
22,17
22,9
16,9
9,9
8,9
13,9
24,9
24,18
20,18
14,18
13,18
10,18
5,18
2,18
2,20
0,20
11,20
18,20
16,20
13,20
15,20
22,20
19,20
19,24
19,14
17,14
14,14
13,14
13,10
22,10
22,19
15,19
9,19
12,19
12,16
3,16
11,16
16,16
16,6
24,6
24,7

b. Number of states expanded=349

A star:

a. First path identified by A star: Cost=10 cells/9 moves

0,0
0,9
0,5
11,5
21,5
20,5
20,15
18,15
24,15
24,7

b. Number of states expanded=360