

Introduction:

Every character in this report and code is written ONLY by myself. The programming language I am using is C++. The code is run and tested on my own computer (Windows x64 platform). I only used standard C++ library, so the code should be able to compile on any platform.

How to compile:

The code directory is as follows

./HW2.sln ...Visual studio 2019 solution file

./HW2 ...The source codes (Timer.h Map.h Map.cpp main.cpp)

./HW2/Debug ...The executable file

If you are on Windows and have Visual studio, open the .sln solution file and hit ctrl+f5 to run or run the executable in ./HW2/Debug directly. If you are on Unix, use g++ to compile without any additional flags.

Algorithm description:

0. Cost Function:

As given in the homework description:

+4 for cleaned up

-1 for move \leftarrow

-1.1 for move \rightarrow

-1.2 for move \uparrow

-1.3 for move \downarrow

-0.2 for suck

0 for NoOp

The smallest cost is the one with highest performance points.

1. Uniform cost tree search:

We construct a fringe as a priority queue. First, expand the initial state and push the result into the fringe. And we enter in a loop. Each time, only the smallest cost node in the fringe is expanded and push the expanded nodes into the fringe. Because it is a priority_queue, the fringe nodes is sorted from highest to lowest according to their performance measure. Then the node is removed from the queue. Each expansion is counted as 1 more depth. If the fringe is empty before expansion, the search will ended as failure.

2. Uniform cost graph search:

3. Depth-limited depth-first tree search:

4. Depth-limited depth-first tree

Result:

Test Case1:

X			X	X
X			X	X
	S			X
X		X		
X			X	X

```
-----Test Case 1-----  
-----Runing Uniform Cost Tree Search-----  
The first 5 nodes to be expand are:  
There are 0 movements in total.  
Cost=0  
There are 1 movements in total.  
- Cost=0  
There are 2 movements in total.  
-- Cost=0  
There are 3 movements in total.  
--- Cost=0  
There are 4 movements in total.  
---- Cost=0  
Total expansion= 72559410  
  
The best performance within 10 movements is:  
There are 10 movements in total.  
->->->+<-+<-+ Cost=8.5  
Finished in 63.984 seconds.
```

X			X	X
X			X	X
	S			X
X		X		
X			X	X

It works as expected because according to the algorithm, it always expand the nodes with a highest performance. In this situation, No-op will produce the highest performance of 0, and all other options will produce a negative performance. But because this is a complete

algorithm, the result is optimal.

For uniform cost graph search:

```
-----Running Uniform Cost Graph Search-----
The first 5 nodes to be expand are:
There are 0 movements in total.
Cost=0
There are 1 movements in total.
← Cost=-1
There are 1 movements in total.
→ Cost=-1.1
There are 1 movements in total.
↑ Cost=-1.2
There are 1 movements in total.
↓ Cost=-1.3
Total expansion= 2365

The best performance within 10 movements is:
There are 10 movements in total.
→→→↑↑←←↑↑ Cost=8.5
Finished in 0.048 seconds.
```

X			X	X
X			X	X
	S			X
X		X		
X			X	X

It gave the same result as uniform cost tree search, but it is much more efficient.

For depth-first tree search:

```
-----Running Depth First Tree Search-----
The first 5 nodes to be expand are:
There are 0 movements in total.
Cost=0
There are 1 movements in total.
- Cost=0
There are 2 movements in total.
-- Cost=0
There are 3 movements in total.
--- Cost=0
There are 4 movements in total.
---- Cost=0
Total expansion= 72559410

The best performance within 10 movements is:
There are 10 movements in total.
↑→→→↑↑←← Cost=8.5
Finished in 48.753 seconds.
```

X			X	X
X			X	X
	S			X
X		X		
X			X	X

It expands one branch of the search tree all the way down. Because the ties are broken randomly, the results will be different every time the program is run. It is also optimal.

For depth-first graph search:

```
-----Running Depth First Graph Search-----
The first 5 nodes to be expand are:
There are 0 movements in total.
Cost=0
There are 1 movements in total.
← Cost=-1
There are 2 movements in total.
←↑ Cost=-2.2
There are 3 movements in total.
←↑+ Cost=1.6
There are 4 movements in total.
←↑+→ Cost=0.5
Total expansion= 356

The best performance within 10 movements is:
There are 9 movements in total.
←↑+→→→→→ Cost=4.8
Finished in 0.008 seconds.
```

X			X	X
X			X	X
	S			X
X		X		
X			X	X

It expands one branch of the search tree unless all the nodes has been visited. Unlike depth-first tree search, it could never perform no-op, because it will have been visited. It didn't provide an optimal solution maybe due to a tie-breaking and some of the visited states are removed.

Test Case 2:

Similar to test case 1, only uniform cost graph search can give us the optimal solution.

```

-----Test Case 2-----
-----Runing Uniform Cost Tree Search-----
The first 5 nodes to be expand are:
There are 0 movements in total.
    Cost=0
There are 1 movements in total.
    Cost=0
There are 2 movements in total.
    Cost=0
There are 3 movements in total.
    Cost=0
There are 4 movements in total.
    Cost=0
Total expansion= 72559410

The best performance within 10 movements is:
There are 10 movements in total.
->+↑+↑←+←←+ Cost=8.7
Finished in 64.038 seconds.
-----Runing Uniform Cost Graph Search-----
The first 5 nodes to be expand are:
There are 0 movements in total.
    Cost=0
There are 1 movements in total.
    ← Cost=-1
There are 1 movements in total.
    → Cost=-1.1
There are 2 movements in total.
    →+ Cost=2.7
There are 3 movements in total.
    →+← Cost=1.7
Total expansion= 1787

The best performance within 10 movements is:
There are 10 movements in total.
->+↑+↑←+←←+ Cost=8.7
Finished in 0.039 seconds.
-----Runing Depth First Tree Search-----
The first 5 nodes to be expand are:
There are 0 movements in total.
    Cost=0
There are 1 movements in total.
    ↑ Cost=-1.2
There are 2 movements in total.
    ↑↑ Cost=-2.4
There are 3 movements in total.
    ↑↑↑ Cost=-3.6
There are 4 movements in total.
    ↑↑↑↑ Cost=-4.8
Total expansion= 72559410

The best performance within 10 movements is:
There are 10 movements in total.
->+↑+↑←+←←+ Cost=8.7
Finished in 49.145 seconds.
-----Runing Depth First Graph Search-----
The first 5 nodes to be expand are:
There are 0 movements in total.
    Cost=0
There are 1 movements in total.
    ← Cost=-1
There are 2 movements in total.
    ←← Cost=-2
There are 3 movements in total.
    ←←↓ Cost=-3.3
There are 4 movements in total.
    ←←↓↓ Cost=-4.6
Total expansion= 624

The best performance within 10 movements is:
There are 10 movements in total.
->+↓+←←←+↑+ Cost=8.6
Finished in 0.012 seconds.

```

Copied from the console (in case of letter encoding issues)

```

-----Test Case 1-----
-----Runing Uniform Cost Tree Search-----
The first 5 nodes to be expand are:
There are 0 movements in total.

```

Cost=0
 There are 1 movements in total.
 - Cost=0
 There are 2 movements in total.
 -- Cost=0
 There are 3 movements in total.
 --- Cost=0
 There are 4 movements in total.
 ---- Cost=0
 Total expansion= 72559410

The best performance within 10 movements is:

There are 10 movements in total.
 →→→+ ↑ +←+ ↑ + Cost=8.5

Finished in 64.38 seconds.

-----Runing Uniform Cost Graph Search-----

The first 5 nodes to be expand are:

There are 0 movements in total.

Cost=0

There are 1 movements in total.

← Cost=-1

There are 1 movements in total.

→ Cost=-1.1

There are 1 movements in total.

↑ Cost=-1.2

There are 1 movements in total.

↓ Cost=-1.3

Total expansion= 2365

The best performance within 10 movements is:

There are 10 movements in total.
 →→→+ ↑ +←+ ↑ + Cost=8.5

Finished in 0.043 seconds.

-----Runing Depth First Tree Search-----

The first 5 nodes to be expand are:

There are 0 movements in total.

Cost=0

There are 1 movements in total.

+ Cost=-0.2

There are 2 movements in total.

++ Cost=-0.4

There are 3 movements in total.

+++ Cost=-0.6

There are 4 movements in total.

++++ Cost=-0.8
Total expansion= 72559410

The best performance within 10 movements is:

There are 10 movements in total.

↑→→+→+↑+←+ Cost=8.5

Finished in 48.989 seconds.

-----Runing Depth First Graph Search-----

The first 5 nodes to be expand are:

There are 0 movements in total.

Cost=0

There are 1 movements in total.

← Cost=-1

There are 2 movements in total.

←↑ Cost=-2.2

There are 3 movements in total.

←↑+ Cost=1.6

There are 4 movements in total.

←↑+→ Cost=0.5

Total expansion= 356

The best performance within 10 movements is:

There are 9 movements in total.

←↑+→→→+→+ Cost=4.8

Finished in 0.008 seconds.

-----Test Case 2-----

-----Runing Uniform Cost Tree Search-----

The first 5 nodes to be expand are:

There are 0 movements in total.

Cost=0

There are 1 movements in total.

- Cost=0

There are 2 movements in total.

-- Cost=0

There are 3 movements in total.

--- Cost=0

There are 4 movements in total.

---- Cost=0

Total expansion= 72559410

The best performance within 10 movements is:

There are 10 movements in total.

→+ ↑ + ↑ ←+←←+ Cost=8.7

Finished in 64.038 seconds.

-----Runing Uniform Cost Graph Search-----

The first 5 nodes to be expand are:

There are 0 movements in total.

Cost=0

There are 1 movements in total.

← Cost=-1

There are 1 movements in total.

→ Cost=-1.1

There are 2 movements in total.

→+ Cost=2.7

There are 3 movements in total.

→+← Cost=1.7

Total expansion= 1787

The best performance within 10 movements is:

There are 10 movements in total.

→+ ↑ + ↑ ←+←←+ Cost=8.7

Finished in 0.039 seconds.

-----Runing Depth First Tree Search-----

The first 5 nodes to be expand are:

There are 0 movements in total.

Cost=0

There are 1 movements in total.

↑ Cost=-1.2

There are 2 movements in total.

↑ ↑ Cost=-2.4

There are 3 movements in total.

↑ ↑ ↑ Cost=-3.6

There are 4 movements in total.

↑ ↑ ↑ ↑ Cost=-4.8

Total expansion= 72559410

The best performance within 10 movements is:

There are 10 movements in total.

→+ ↑ + ↑ ←+←←+ Cost=8.7

Finished in 49.145 seconds.

-----Runing Depth First Graph Search-----

The first 5 nodes to be expand are:

There are 0 movements in total.

Cost=0

There are 1 movements in total.

← Cost=-1

There are 2 movements in total.

←← Cost=-2

There are 3 movements in total.

←←↓ Cost=-3.3

There are 4 movements in total.

←←↓↓ Cost=-4.6

Total expansion= 624

The best performance within 10 movements is:

There are 10 movements in total.

→+↓+←←←+↑+ Cost=8.6

Finished in 0.012 seconds.