Name: Wenhao Li ID:#14328112

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 ←
- -1.1 for move \rightarrow
- -1.2 for move ↑
- -1.3 for move ↓
- -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:

Similar to uniform cost tree search stated above. The difference is that we created a new array called 'visited'. And every time we pick a node in the fringe, we first compare it with all visited nodes and check whether the node has been visited before. If it is, delete it from the fringe to avoid revisiting.

3. Depth-limited depth-first tree search:

We construct the fringe as a stack (LIFO). First, expand the initial state and push the one of the resulting nodes randomly into the fringe. And we enter in a loop. Each time, we expand the node at the top of the fringe, choose one of the resulting nodes randomly (to break the tie of the nodes at the same depth) and push it into the fringe. Each expansion is counted as 1 depth. If no more expansion is possible, we pop off this node from the fringe.

4. Depth-limited depth-first tree

Similar to depth-limited depth-first tree stated above, the only difference is we created a new array called 'visited'. Each time, we choose the node at the top of the fringe and we compare the resulting expanded nodes with all visited nodes to check whether the node has been visited before. If it is, delete it from the resulting nodes to avoid revisiting.

Result:

The - character in the screenshot represents No-op, + represents suck.

Test Case1:

The initial state is under follows, X representing a dirty room and S for the initial position of the vacuum cleaner.

Χ			Χ	Χ
Χ			Χ	Χ
	S			Χ
Χ		Χ		
Χ			Χ	Χ

For uniform cost tree search, the program produce this result:

```
Runing Uniform Cost Tree Search
   first 5 nodes to be expand are:
      are 0 movements in total.
         Cost=0
        e 1 movements in total.
          Cost=0
        re 2 movements in total.
          Cost=0
         e 3 movements in total.
                                                                                           Χø
          Cost=0
      are 4 movements in total.
                                                                                           XΦ
                                                                                                   ø
                                                                                                          ø
         Cost=0
otal expansion= 72559410
                                                                                                  S+
The best performance within 10 movements is:
                                                                                           χ
                                                                                                         Χø
                                                                                                   Ð
here are 10 movements in total.

\rightarrow \rightarrow + \uparrow + \leftarrow + \uparrow + Cost=8.5
                                                                                           Χø
                                                                                                                 Χø
                                                                                                                        Χø
```

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:

```
The first 5 nodes to be expand are:
There are 0 movements in total.
             Cost=0
         are 1 movements in total.
             Cost =-1
         are 1 movements in total.
             Cost =-1.1
        are 1 movements in total.
Cost=-1.2
                                                                                                                    Χø
                                                                                                                                                         Χø
                                                                                                                              ٥
                                                                                                                                       ٥
         are 1 movements in total.
Cost=-1.3
                                                                                                                    Χø
        expansion= 2365
                                                                                                                             S
The best performance within 10 movements is:
There are 10 movements in total.
→→→→+↑+←+↑+ Cost=8.5
Finished in 0.048 seconds.
                                                                                                                    Χø
                                                                                                                                       Χø
                                                                                                                    Χø
                                                                                                                                                Χø
                                                                                                                                                         Χø
```

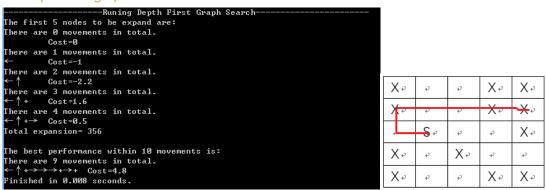
It gave the same result as uniform cost tree search, but it is much more efficient.

For depth-first tree search:

```
-Runing Depth First Tree Search
he first 5 nodes to be expand are:
here are 0 movements in total.
         Cost=Ø
are 2 movements in total.
         Cost=0
                                                                                     Χø
There are 3 movements in total.
         Cost=0
                                                                                     Χø
here are 4 movements in total.
         Cost=0
Total expansion= 72559410
                                                                                            $ 0
                                                                                                                Χø
                                                                                                   ø
The best performance within 10 movements is:
                                                                                     Χø
                                                                                                   Χ
                                                                                                                 ęJ
There are 10 movements in total. \uparrow \rightarrow \rightarrow + \rightarrow + \uparrow + \leftarrow + Cost=8.
                                                                                     Χø
                           Cost =8.5
                                                                                                          Χø
                                                                                                                Χø
Finished in 48.753 seconds
```

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:



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.

```
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:
The best performance within to movements in the first term of the
 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. 	o+ 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. \rightarrow + \uparrow + \uparrow < +< < + Cost=8.
 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. \uparrow \uparrow \uparrow Cost=-3.6
There are 4 movements in total.
↑ ↑ ↑ ↑
Total expansion= 72559410
The best performance within 10 movements is:
There are 10 movements in total.
\rightarrow+\uparrow+\uparrow\leftarrow+\leftarrow+\leftarrow+ Cost=8.7
Finished in 49.145 seconds.
```

```
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.
```

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.
\longrightarrow \longrightarrow +\uparrow +\longleftarrow +\uparrow +
                            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.

$$\uparrow \rightarrow \rightarrow + \rightarrow + \uparrow + \leftarrow +$$
 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.

 \leftarrow ↑ + Cost=1.6

There are 4 movements in total.

 $\leftarrow \uparrow + \rightarrow Cost = 0.5$

Total expansion= 356

The best performance within 10 movements is:

There are 9 movements in total.

$$\leftarrow\uparrow+\rightarrow\rightarrow\rightarrow+\rightarrow+$$
 Cost=4.8

Finished in 0.008 seconds.



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.

```
\rightarrow+ \uparrow + \uparrow \leftarrow+ \leftarrow+
                               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.
\rightarrow+
          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.
\rightarrow+ \uparrow + \uparrow \leftarrow+ \leftarrow+
                               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.
\uparrow \uparrow
          Cost = -2.4
There are 3 movements in total.
\uparrow \uparrow \uparrow Cost=-3.6
There are 4 movements in total.
\uparrow \uparrow \uparrow \uparrow
                    Cost = -4.8
Total expansion= 72559410
The best performance within 10 movements is:
There are 10 movements in total.
\rightarrow+ \uparrow + \uparrow \leftarrow+ \leftarrow+
                               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.

There are 3 movements in total.

$$\leftarrow$$
 Cost=-3.3

There are 4 movements in total.

Total expansion= 624

The best performance within 10 movements is:

There are 10 movements in total.

$$\rightarrow$$
+ \downarrow + \leftarrow + \uparrow + Cost=8.6

Finished in 0.012 seconds.