Hunter Hawkins Stark
05/29/2020
Project #1

A search problem is a problem that appears in many different computational problems. I would say almost all problems have some sort of search embedded in them. The two main categories of searches are uninformed and informed, and the difference lies in an uninformed search where there is no knowledge of the goal other then the initial prompt, and an informed search where there is more knowledge of the goal and how to get there which is called a heuristic. For every search problem there are a couple defining characteristics. There is a start state where the problem is initially defined, the state space which is the set of all states, and the goal state where the problem is supposed to end. With many different search problems there are also many different algorithms that try to solve these problems.

Here is the initial prompt given to the user on how to run the program and what each symbol means. I attached it here to assist in the grading process and make it more user friendly. I have never programmed in python before this class, so it was an interesting challenge to tackle.

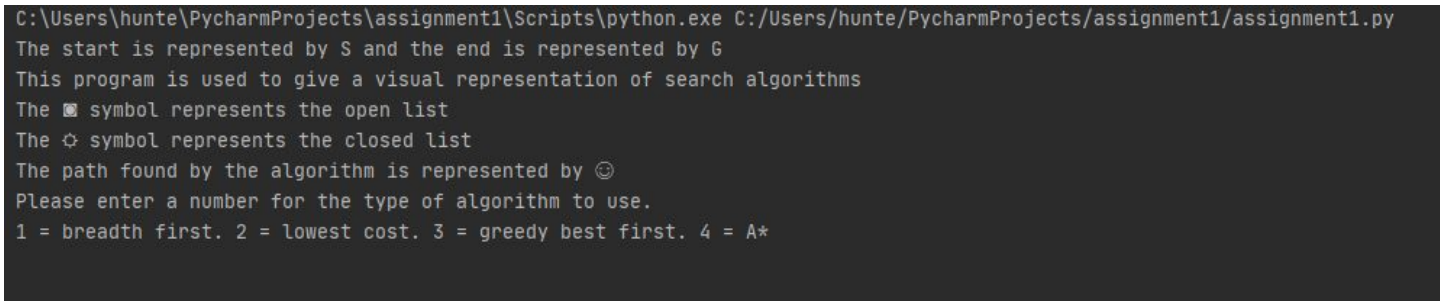
S - The character S represents the start

G - The character G represents the end

☺ - The smiley face represents the path taken

◙ - The square with the filled in circle represented the open list

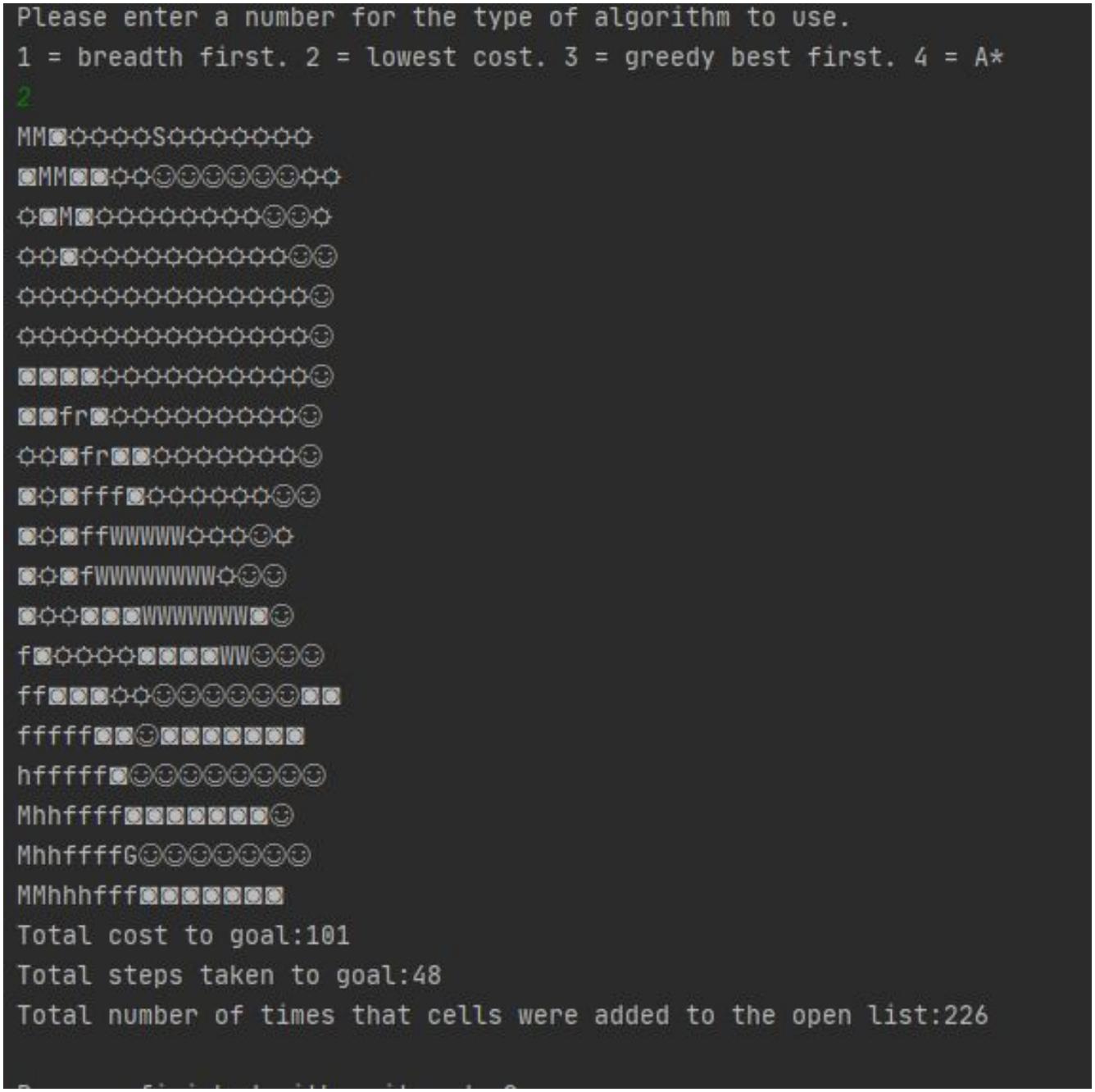
☼ - The sun (i think that's what it is) represents the closed list



Breadth First Search:

```
Please enter a number for the type of algorithm to use.
1 = breadth first. 2 = lowest cost. 3 = greedy best first. 4 = A*
1
○○☺☺☺☺☺S○○○○○○■f
○○☺○○○○○○○○○○■ff
○○☺○○○○○○○○○○■fff
○○☺○○○○○○○○○○■Fff
○○☺○○○○○○○○○○■FFF
○○☺○○○○○○○○○○■fff
○○☺○○○○○○○○○○■fff
○○☺○○○○○○○○○○■fff
○○☺○○○○○○○○○○■fff
○○☺○○○○○○○○○○■Fff
○○☺○○○WWWWW○■FFF
○○☺○○WWWWWWWWFFF
○○☺○○○○WWWWWWWrr
○○☺○○○○○○■fWWfff
○○☺○○○○○○■ffffff
○○☺○○○○○○■ffffff
○○☺○○○○○○■RRRRRR
○○☺○○○○○○■ffffff
○○☺☺☺☺☺☺Gfffffff
○○○○○○○○■fffffff
Total cost to goal:108
Total steps taken to goal:28
Total number of times that cells were added to the open list:200

Process finished with exit code 0
```

Lowest Cost:


Please enter a number for the type of algorithm to use.
1 = breadth first. 2 = lowest cost. 3 = greedy best first. 4 = A*
2
Total cost to goal:101
Total steps taken to goal:48
Total number of times that cells were added to the open list:226

Greedy Best First With Euclidean Distance:

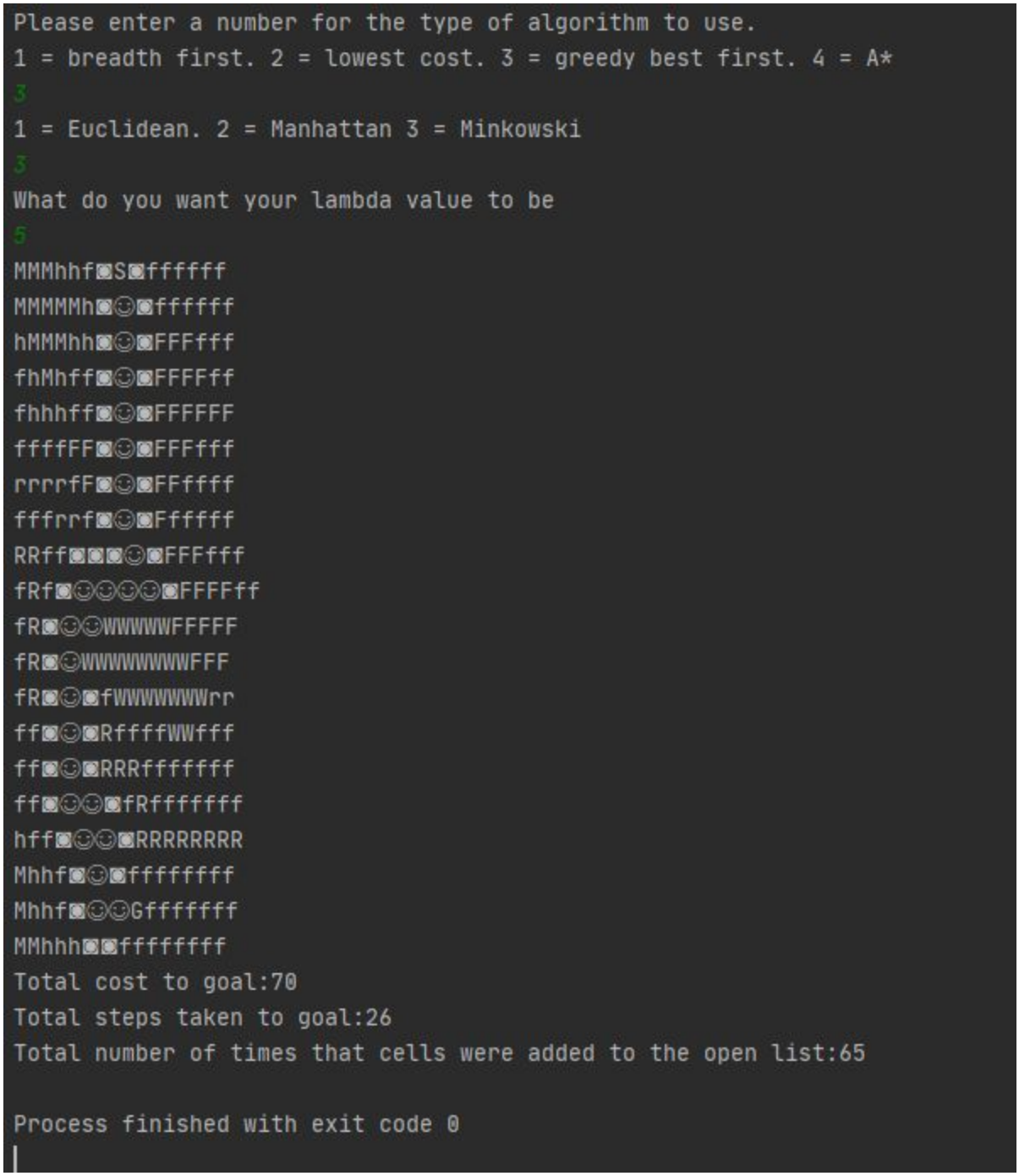
```
Please enter a number for the type of algorithm to use.
1 = breadth first. 2 = lowest cost. 3 = greedy best first. 4 = A*
3
1 = Euclidean. 2 = Manhattan 3 = Minkowski
1
MMMhhf◙S◙ffffff
MMMMMh◙☺◙ffffff
hMMMhh◙☺◙FFFfff
fhMhff◙☺◙FFFFff
fhhhff◙☺◙FFFFFF
ffffFF◙☺◙FFFfff
rrrrfF◙☺◙FFffff
fffrrf◙☺◙Ffffff
RRff◙◙☼☺☼◙FFfff
fRf◙☺☺☺☺☼☼◙FFff
fR◙☺☺WWWWWFFFFF
fR◙☺WWWWWWWWFFF
fR◙☺◙fWWWWWWWrr
ff◙☺◙RffffWWfff
ff◙☺◙RRRfffffff
ff◙☺◙ffRfffffff
hf◙☺◙ffRRRRRRRR
Mh◙☺◙◙◙◙ffffffff
Mh◙☺☺☺☺Gfffffff
MMh◙◙◙◙ffffffff
Total cost to goal:70
Total steps taken to goal:26
Total number of times that cells were added to the open list:72

Process finished with exit code 0
```

Greedy Best First With Manhattan Distance:

```
Please enter a number for the type of algorithm to use.
1 = breadth first. 2 = lowest cost. 3 = greedy best first. 4 = A*
3
1 = Euclidean. 2 = Manhattan 3 = Minkowski
2
MMMhhf■S■ffffff
MMMMMh■☺■ffffff
hMMMhh■☺■FFFfff
fhMhff■☺■FFFFff
fhhhff■☺■FFFFFF
ffffFF■☺■FFFfff
rrrrfF■☺■FFfffff
fffrrf■☺■Ffffff
RRff■■☼☺☼■FFfff
fRf■☺☺☺☺☼☼■FFff
fR■☺☺WWWWWFFFFF
fR■☺WWWWWWWWFFF
fR■☺■fWWWWWWWrr
ff■☺■RffffWWfff
ff■☺■RRRfffffff
ff■☺■ffRfffffff
hf■☺■ffRRRRRRRR
Mh■☺■■■■ffffffff
Mh■☺☺☺☺Gfffffff
MMh■■■■fffffffff
Total cost to goal:70
Total steps taken to goal:26
Total number of times that cells were added to the open list:72

Process finished with exit code 0
```

I noticed with Greedy best first that the Manhattan distance and the euclidean distance generally returned the similar if not the same output for most inputs. The larger the graph gets the more varied their respective outputs get.

Greedy Best First with Minkowski Distance:

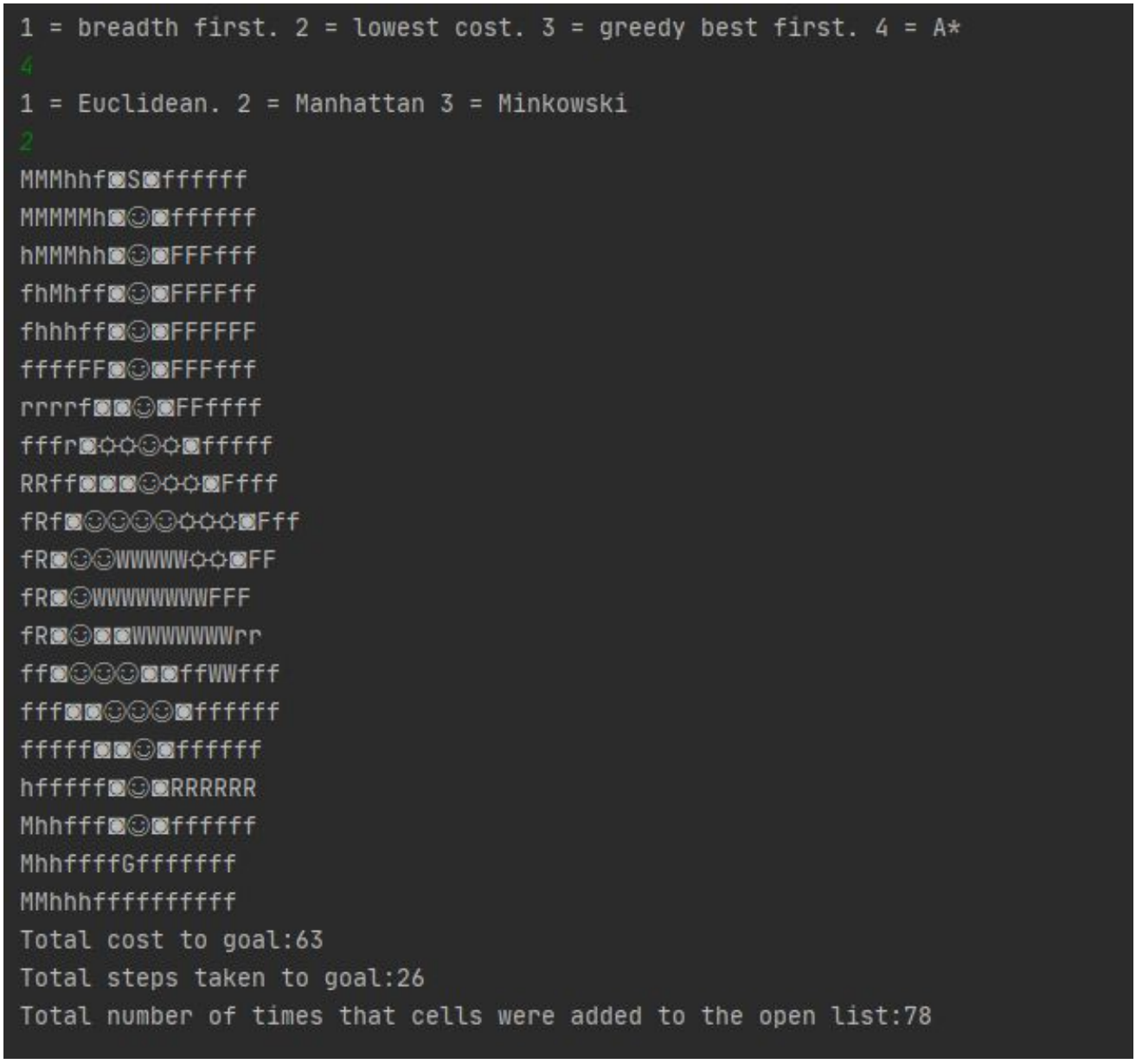


I never heard of the Minkowski distance prior to this program so I thought it was an interesting one to include in there. The lambda value seems to have a big impact on the efficiency of the search.

A* With Euclidean Distance:

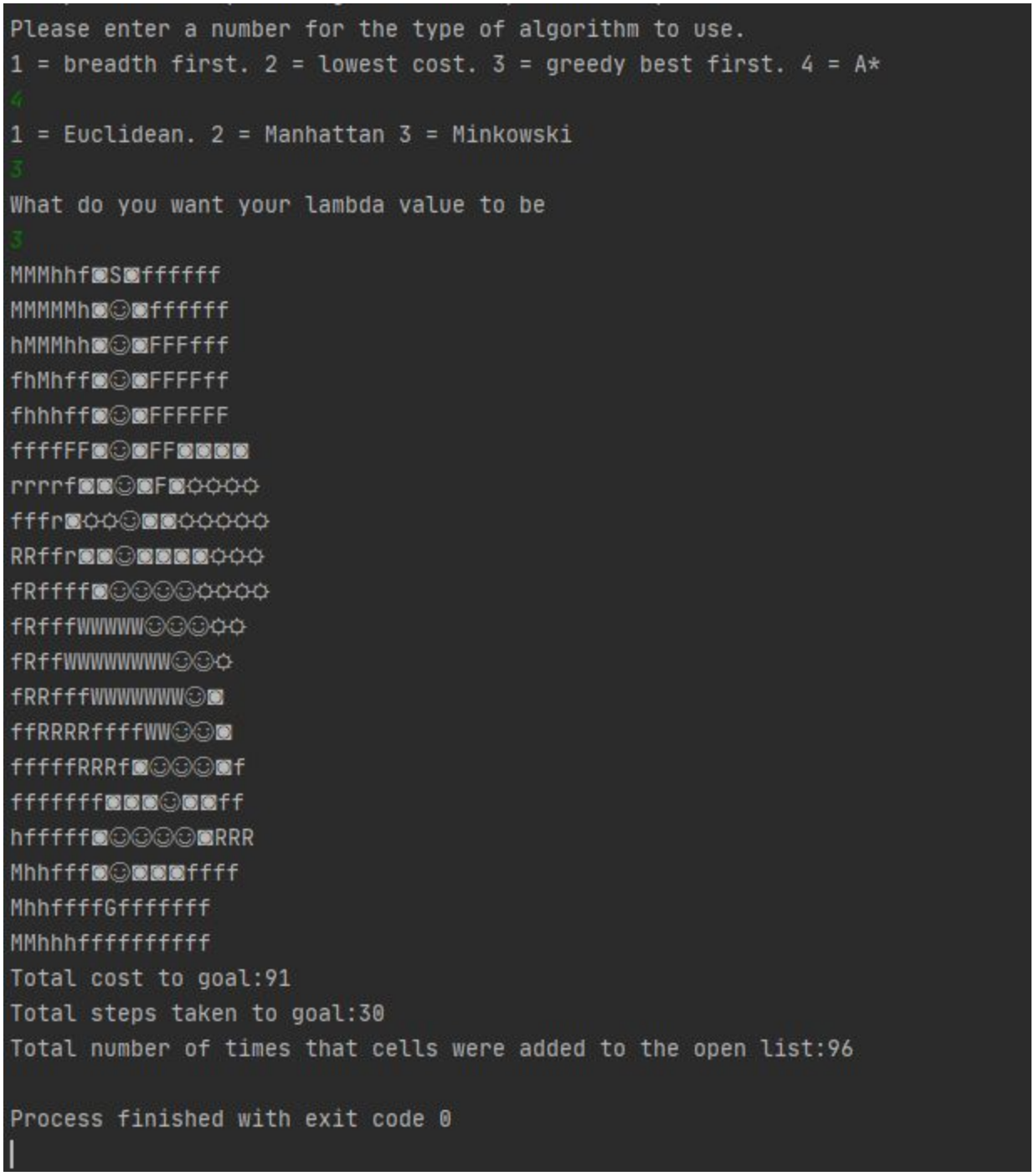
```
1 = breadth first. 2 = lowest cost. 3 = greedy best first. 4 = A*
4
1 = Euclidean. 2 = Manhattan 3 = Minkowski
1
MMMhhf■S■ffffff
MMMMMh■☺■ffffff
hMMMhh■☺■FFFfff
fhMhff■☺■FFFFff
fhhhff■☺■FFFFFF
ffffFF■☺■FFFfff
rrrrf■■☺■FFfffff
fffr■✿✿☺✿■fffff
RRff■■■☺✿✿■Ffff
fRf■☺☺☺☺✿✿✿■Fff
fR■☺☺WWWWW✿✿■FF
fR■☺WWWWWWWWFFF
fR■☺■■WWWWWWWrr
ff■☺☺☺■■ffWWfff
fff■■☺☺☺■ffffff
fffff■■☺■ffffff
hffffff■☺■RRRRRR
Mhhfff■☺■ffffff
MhhffffGfffffff
MMhhhfffffffffff
Total cost to goal:63
Total steps taken to goal:26
Total number of times that cells were added to the open list:78

Process finished with exit code 0
```

A* With Manhattan Distance:



As with the above greedy algorithms I did not see a huge variation in the results between the A* with Euclidean and A* with Manhattan distance. As the graph grew in size I found slight differences but nothing substantial.

A* with Minkowski Distance:



Conclusion:

Overall I feel my results were consistent with what I was expecting. The search that I would say performed the best overall would be A* with Manhattan or Euclidean distance. Those two searches took a minimum total cost to the goal and a minimum total steps taken to the goal. The greedy best first had the least amount of items added to the open list which makes since due to it taking a pretty direct path. The lowest cost search had the most total steps taken to the goal

and had the most items on the open list. The breadth first search had the highest total cost to goal and also had a lot of items on the open list. I believe these findings line up with what we learned about the algorithms previously and I personally enjoyed learning and coding the A* algorithm as it seems like an interesting search. I believe all the algorithms worked as desired, unless I am misinterpreting how they are supposed to work. The overall project has been beneficial to my learning by not only starting to learn python but also learning new search algorithms.