AI Project 1 Report
Team: Hazam

ESLAM ELSHARKAWY 201800473
MOHAMED ISMAIL 201800595
MOHAMED HAZEM 201802190

Workload Distribution

Mohamed Ismail:

1. Problem Formulation and Modeling

Eslam El Sharkawy:

1. Informed Search Algorithms
    a. A*
    b. GFS
    c. Hill Climbing
    d. Local Beam Search
    e. Simulated Annealing

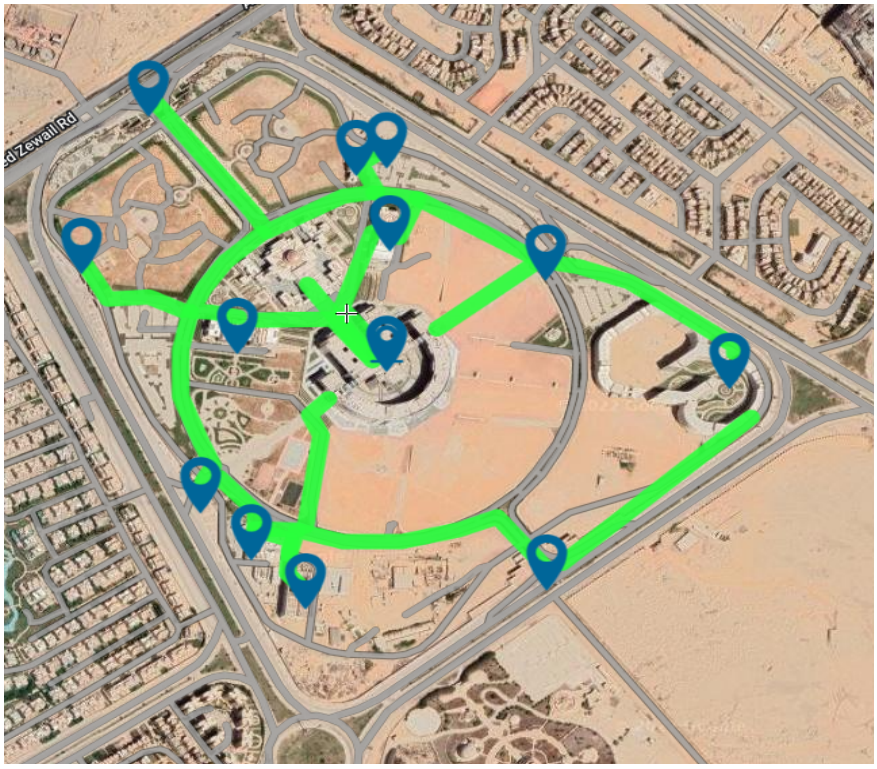
Mohamed Hazem:

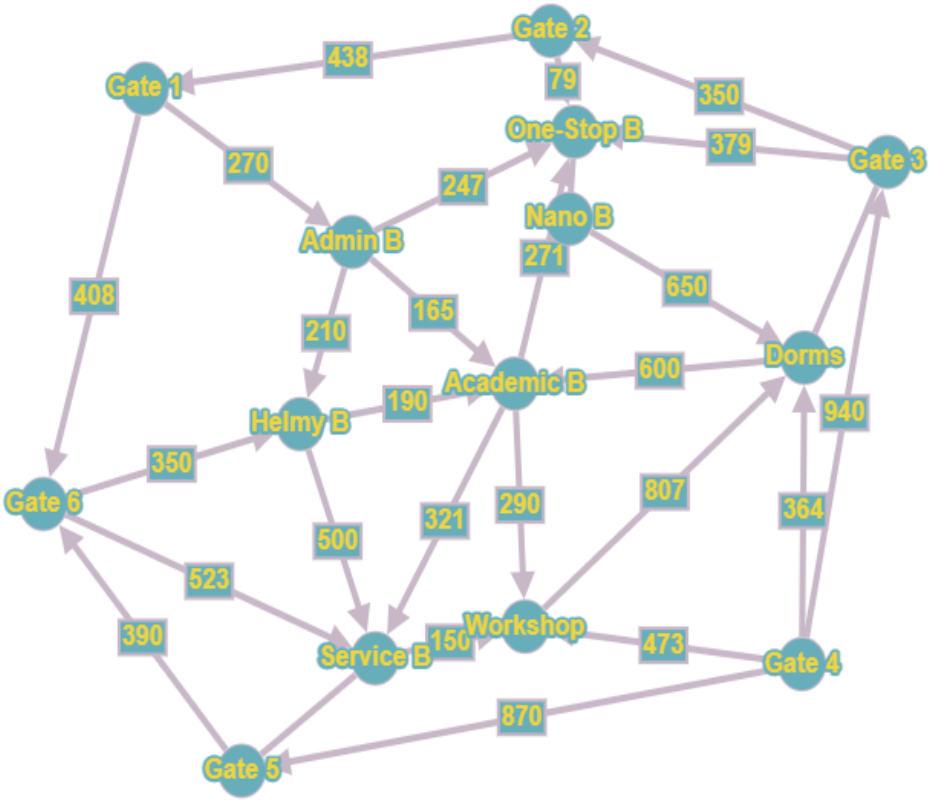
1. Uninformed Search Algorithms
    a. BFS
    b. DFS
    c. IDS
    d. UCS

Problem Formulation

We solved the problem using a maze approach.
First we started by extracting multiple images for zewail city from google maps

Then we marked the landmarks and extracted x, y location relative to the center of the city. These x, y values are used for the euclidean heuristics.


Ahmed Zewail Rd


Gate 2
438
Gate 1
79
350
One-Stop B
379
Gate 3
270
247
Nano B
Admin B
271
650
408
165
210
Dorms
600
Academic B
190
940
Helmy B
350
Gate 6
807
290
321
364
500
523
390
Workshop
150
473
Service B
Gate 4
870
Gate 5

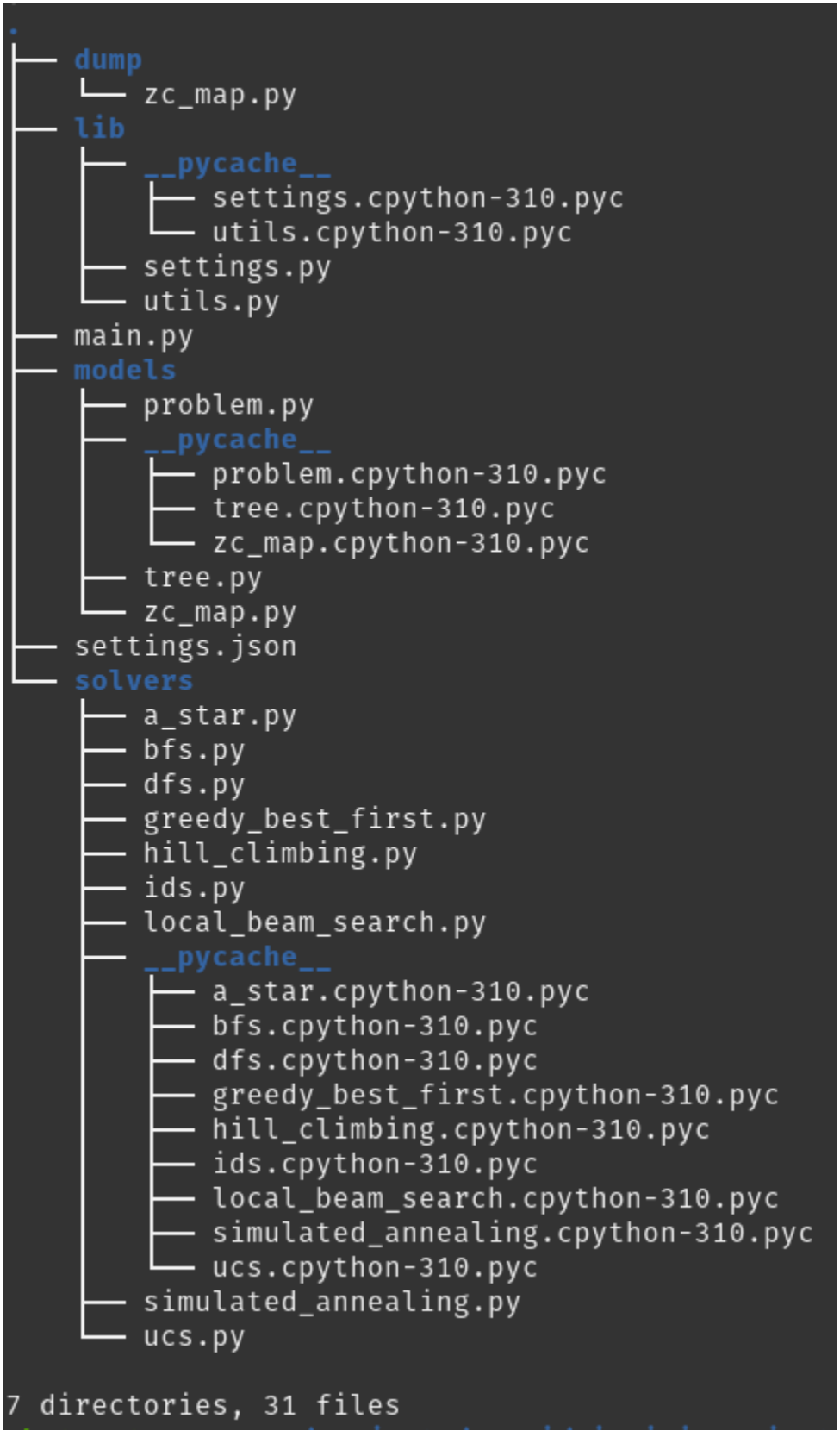
After that we divided the city map into a grid. Each grid cell is 20m squared.
Then we converted this grid to ascii where symbol **#** is a wall or unreachable area, the symbol **=** is a car road, the symbol **-** is passenger road, **x** is a car crossing and **$** is a car slow down. And this is the final result

```
_ZC_ROAD_MAP = '''
##############################
############A#################
###====$$==========$$======###
###=#######==#######===#####
###==#######==########=######
###=#######$$########=######
#F==########==#####-----H---B##
##==########==#####I---------##
##==########==#####----------##
##==###-====xx=====---#####==##
##$$-#--====G======---#####$$##
##==-#--=--#####--=---#####==##
##==-#--L--#####--K=--#####==##
#Exx-#==##-#####-##==#####xxC#
#===----==##-#####-##==#####==-#
#$$=-#=======J========#####$$-#
#===-#--===#######===---#####==-#
#==M-#--===#####==========-##==-#
#==--======#####==========P==-#
#==N-#--==========---####-==-#
#==----------=-----######-==-#
#==---###----xx---########-==-#
#==--#####==================#
############D#################
##############################
'''
```

And these are the step costs for all reachable areas

```
step_cost_map = {
          '=': 1,  # cars road
          '-': 2,  # passengers road
          'x': 3,  # crossing
          '$': 4,  # car slow down
      }
```

Project Structure

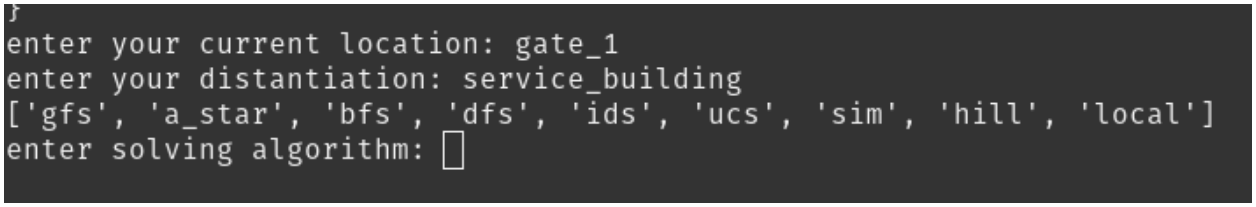


Showcasing Results

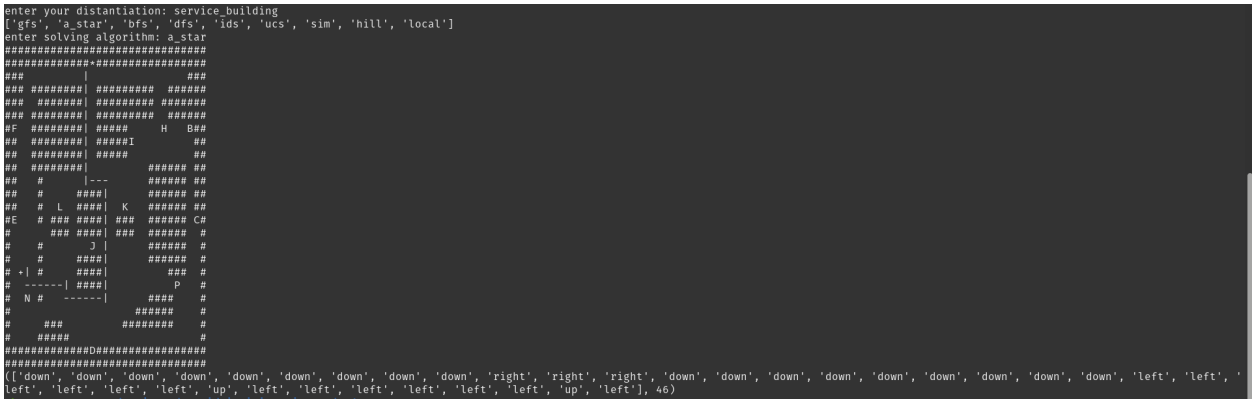
You start the program by running the main.py file.

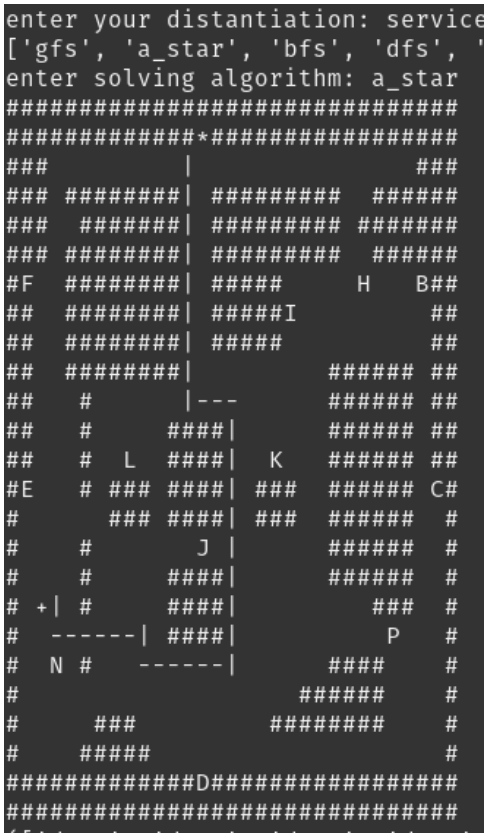
This is the prompt that appears when you start the program.
First it asks for your current location. You can enter a specific zewail city landmark or you can enter general x, y in this form y,x.

After that we have the algorithms prompt.

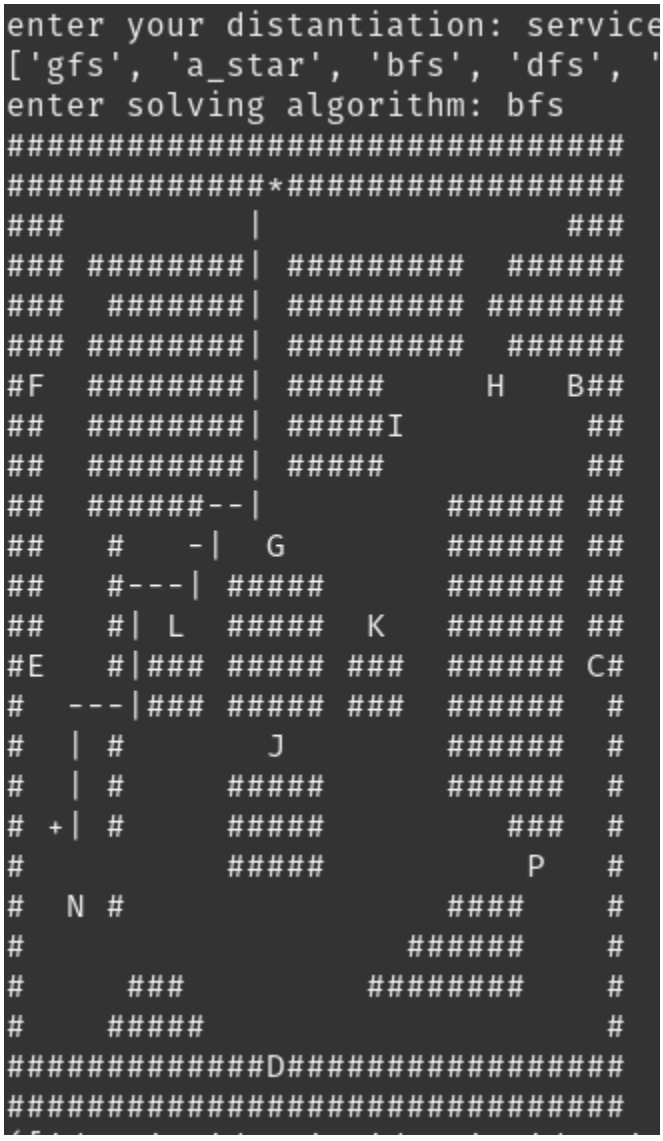


After we choose a certain algorithm for example a_star. The solution appears if it exists and the program draws a map to follow to reach your destination.



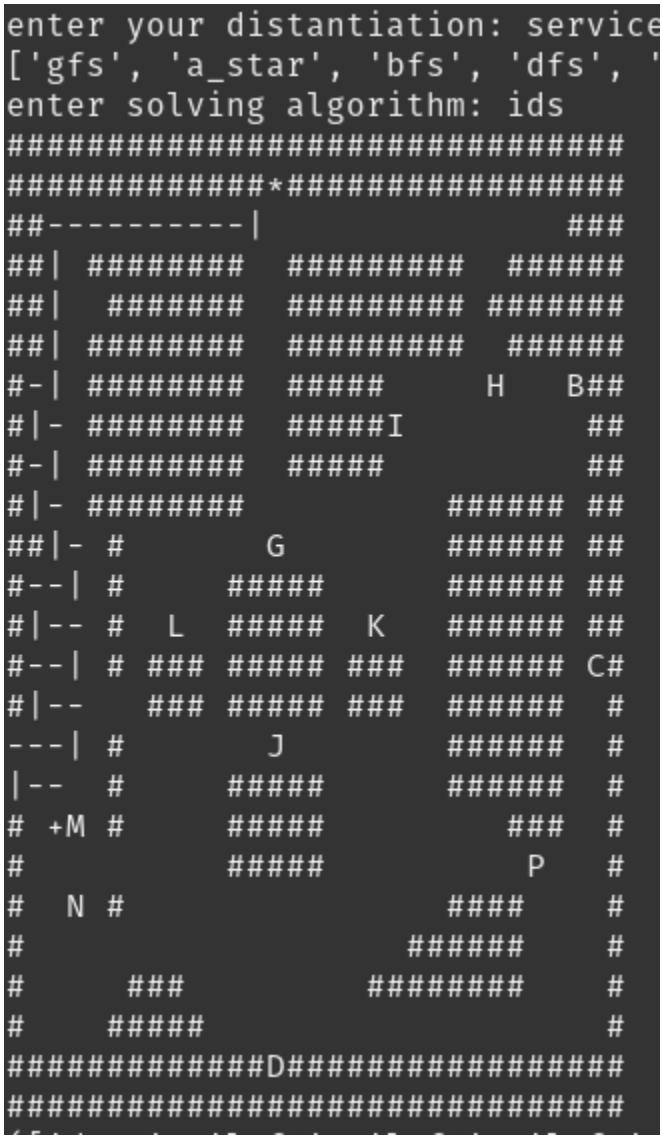


BFS Test



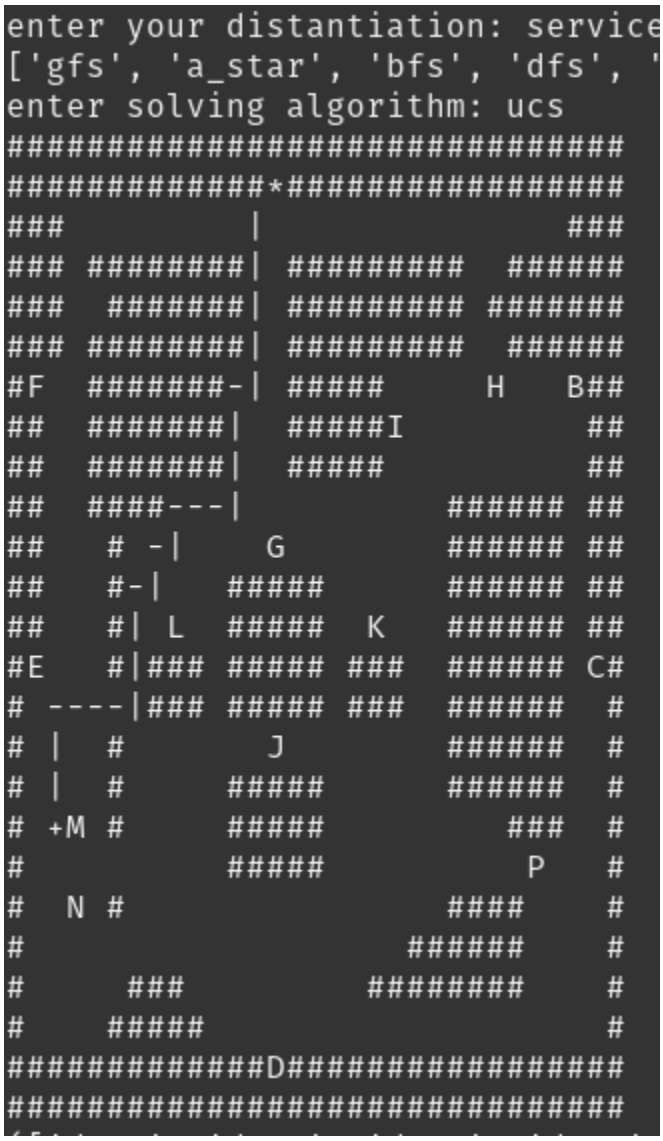
Path cost is 40 (Not Optimal)

IDS



Path Cost is 68 (Not Optimal)

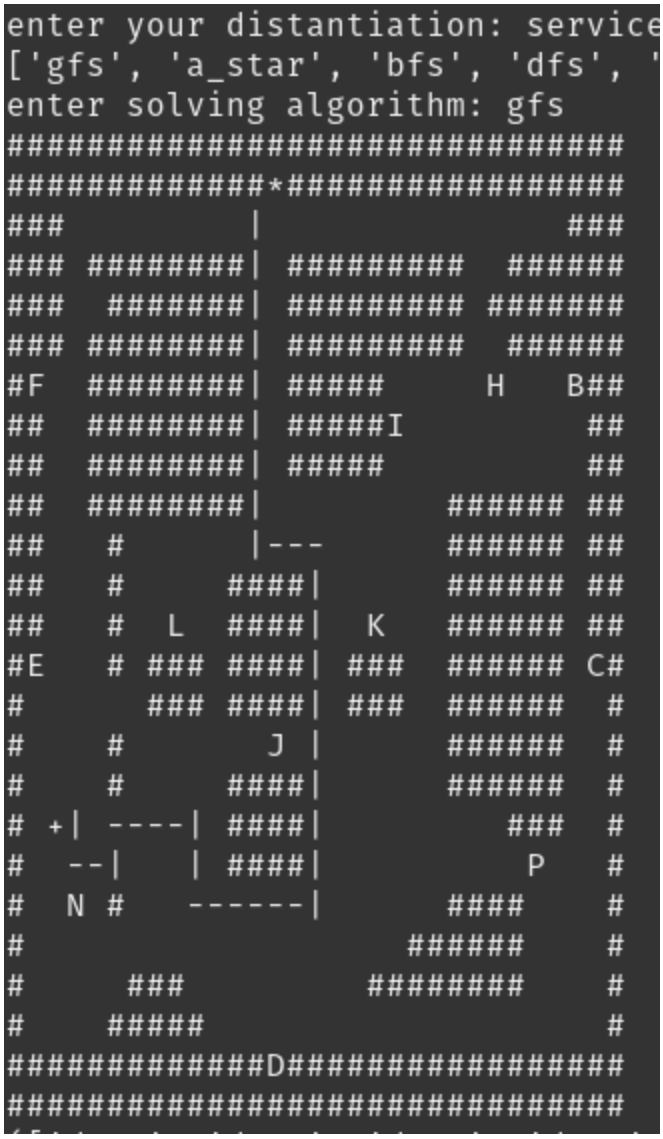
UCS



Path Cost is 38 (Optimal)

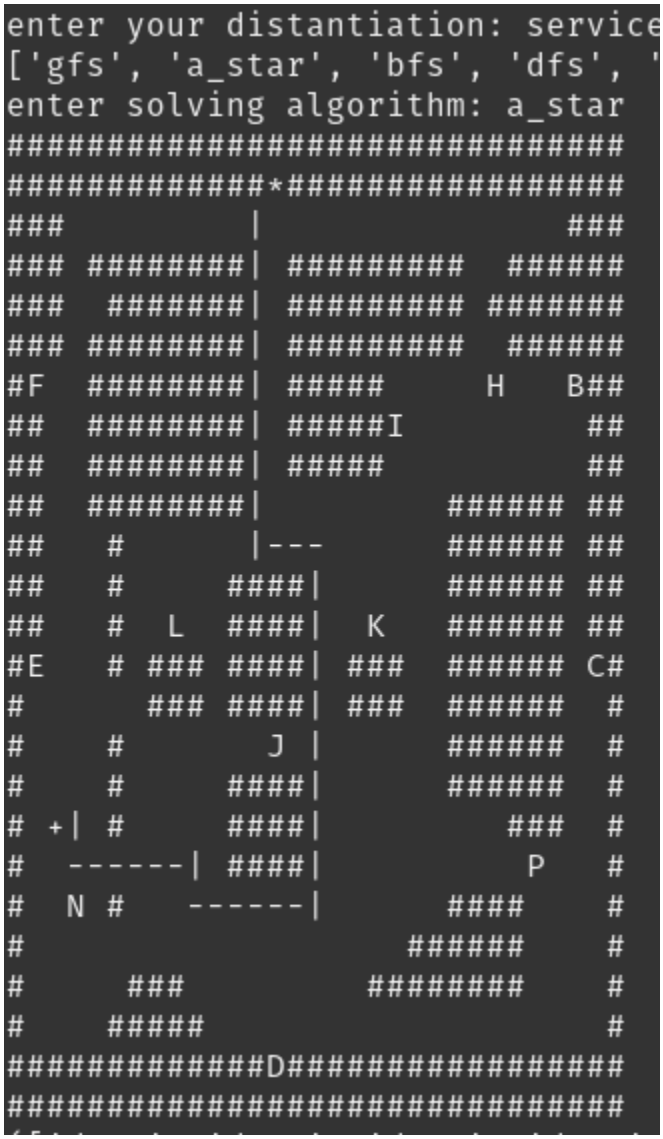
DFS

Always Stuck. We think that is because the complex nature of the map



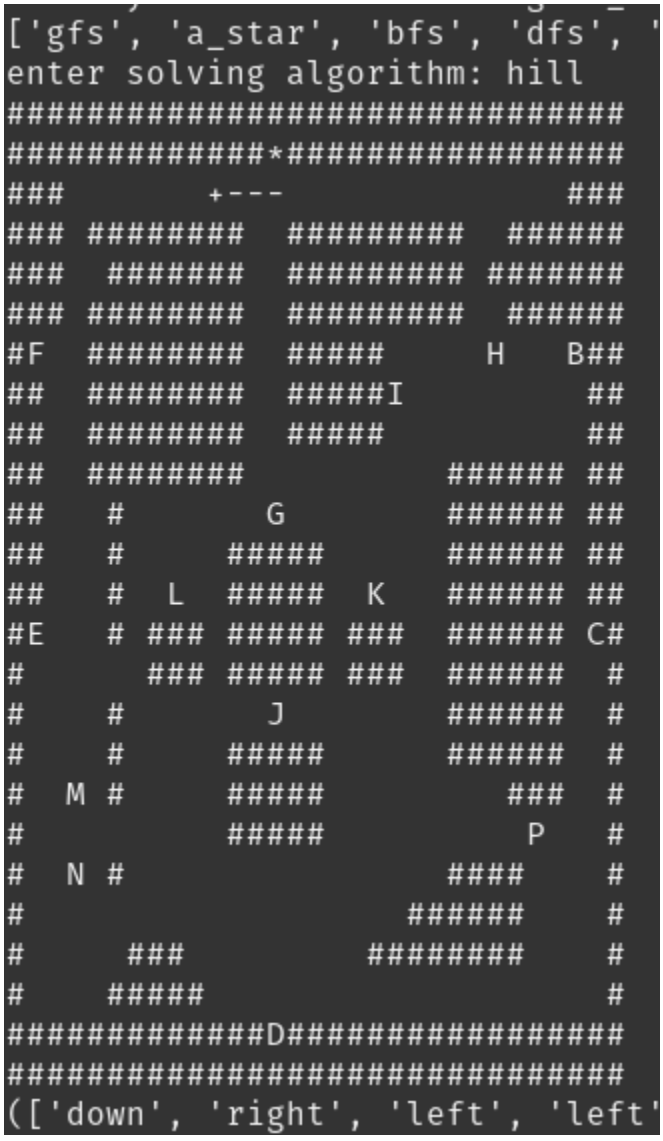
Path Cost 50 (Not Optimal)

A*



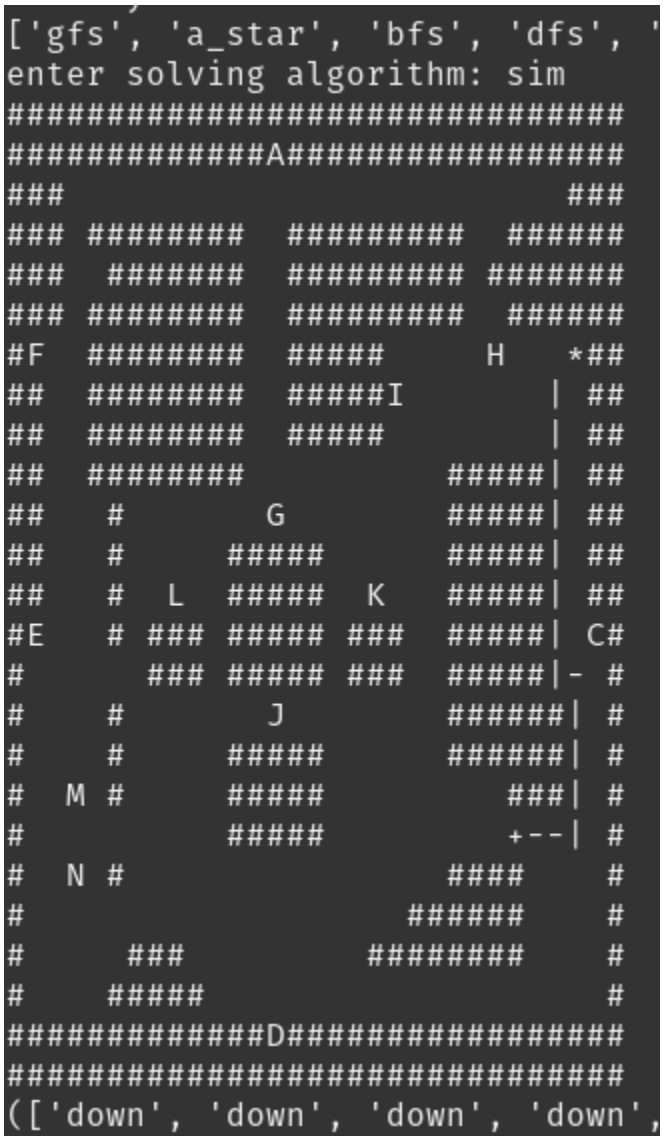
Path Cost 46 (Optimal)

Hill Climbing



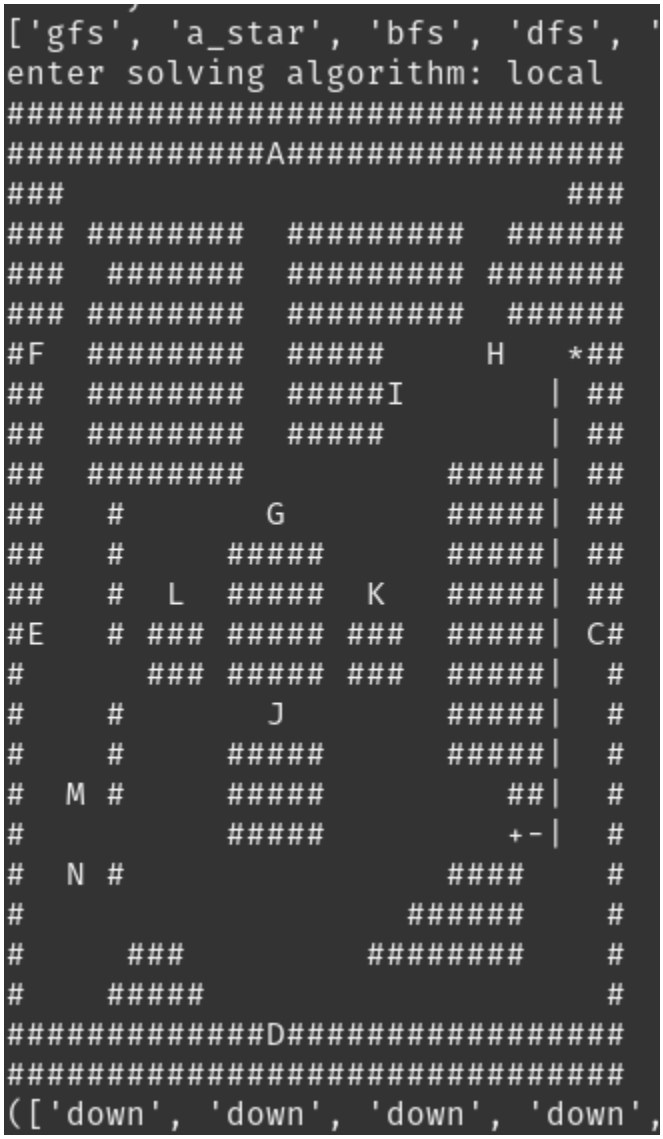
Always stuck in a local minima due to car slow downs

Simulated Annealing



Path Cost 27 (Not Optimal)

Local Beam search



Path Cost 23 (Not Optimal)