Greedy Algorithms

Vasanth Reddy

CS 5114

Content

- □ Introduction
- □ Algorithms
- □ Iterative Depth Search
- □ Greedy Best First Search
- □ epsilon Greedy Best First Search
- □ Problem, Inputs, Objective
- □ Results
- □ Conclusion

Introduction

- □ Search algorithm is any algorithm which solves the search problem, namely, to retrieve information stored within some data structure
- □ Search to find a desired variable in a graph/tree
- □ Vehicle Routing Problem
- □ Project is mainly based on finding a goal state from a start state; i.e Optimal cost problem

Algorithms

□ **Un-informed/Naive/Brute-force Search**: It is a naive search method.

- ➔ No information about path cost or number of steps
- ➔ They just differentiate present state and desired state
- ➔ Iterative Deepening Search(IDS) is explored in this project

□ **Informed Search**

- ➔ These algorithms are little cognitive about the path they choose
- ➔ Greedy Best First Search and epsilon Greedy Best First Search algorithms are explored.

Iterative Deepening Search

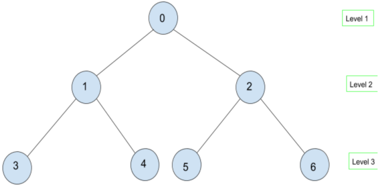
□ Naive approach

□ It is the combination of BFS and DFS.

□ IDS calls DFS for different depths starting from an initial value. In every call, DFS is restricted from going beyond given depth

□ It is like DFS in BFS fashion

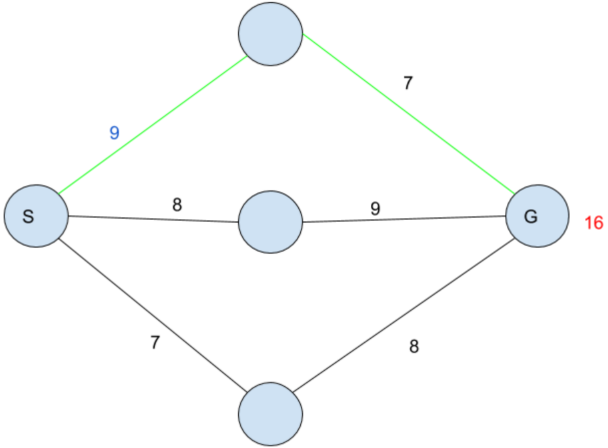
□ Complexity is $O(b^d)$, where b is the number of children nodes and d is the depth of the tree/graph



Depth	Search Nodes
1	0
2	0, 1, 2
3	0, 1, 3, 4, 2, 5, 6

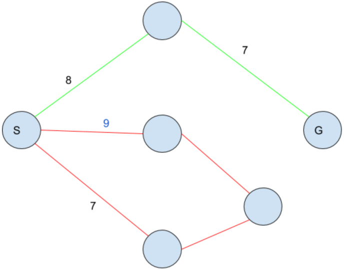
Greedy Best First Search

- □ The idea of informed search is to choose adjacent node with a given heuristic function
- □ In Greedy Best First Search, the heuristic function $h(n) =$ select the node with cheapest cost
- □ It used the above heuristic function to expand
- □ Time Complexity is $O(n * logn)$


7
9
S
8
9
G
16
7
8

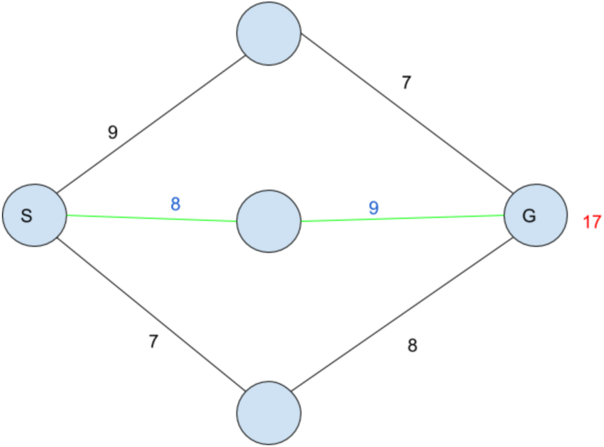
□ It is not optimal sometimes

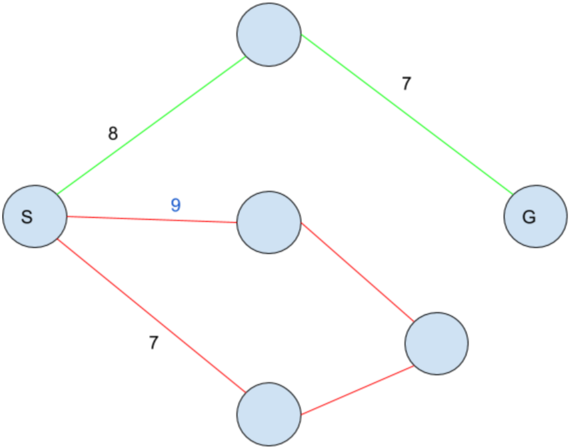
□ It get stuck in loop and doesn't reach goal state.



epsilon Greedy Best First Search

- □ Inspired to develop this algorithm from epsilon feature used in Reinforcement Learning algorithms.
- □ It is same as the Greedy Best First Search but with a little modification
- □ In order to overcome the problems associated GBFS some features have been added
- □ Randomness have been added to Greedy Search.
- □ Algorithm is epsilon times random otherwise it'll be greedy
- □ Time Complexity is $O(n * logn)$


9
7
8
9
S
G
17
7
8

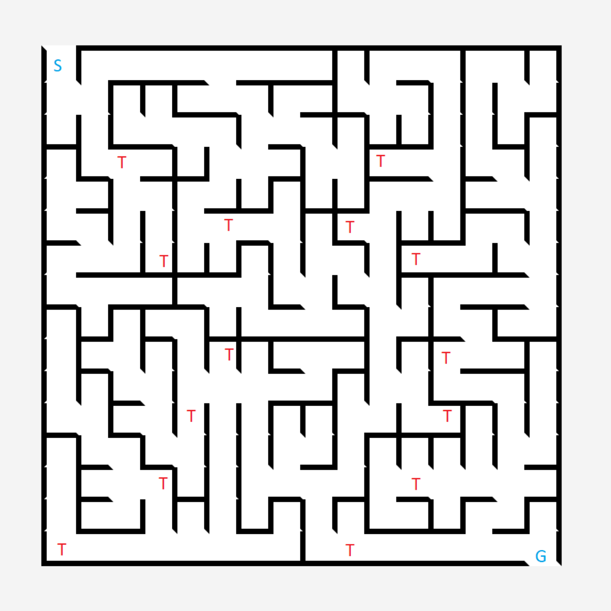

8
7
S
9
G
7

Problem, Inputs, Objective

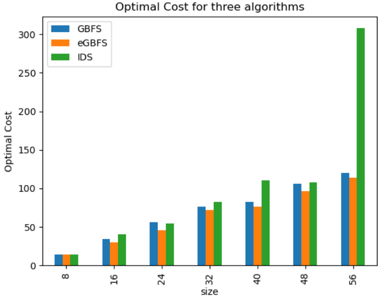
□ A maze is a path or collection of paths, typically from an entrance to a goal. The main objective for any solver is to get to the terminal state/goal state from a given point

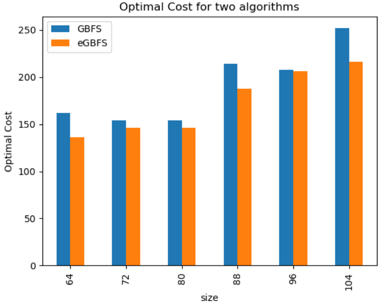
□ Generate the Maze randomly with the following inputs.

- ➔ Maze Size; This gives the dimension of the maze. Contains in the format of rows and columns in a given matrix
- ➔ Wall Locations
- ➔ Trap Locations; Both the wall and trap locations are randomly generated and are randomly distributed throughout the maze. Solver will be penalised for getting into trap which would ultimately result in increase of cost.
- ➔ Goal Locations
- ➔ Start Location

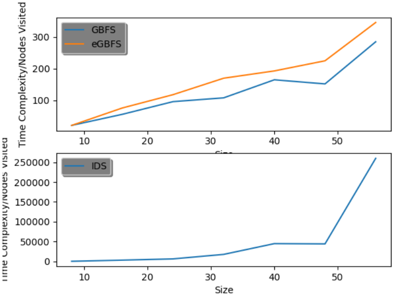

S
T
T
T
T
T
T
T
T
T
T
T
T
T
G

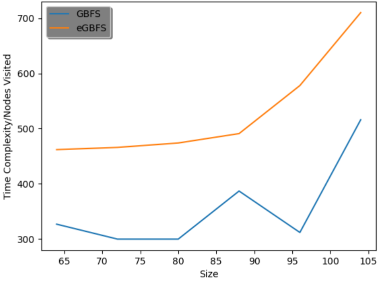
Experiment and Results



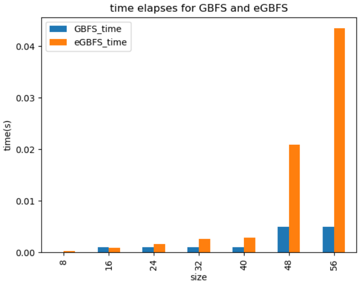


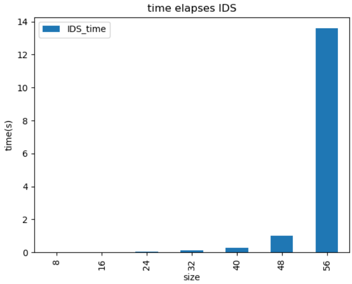
Complexity

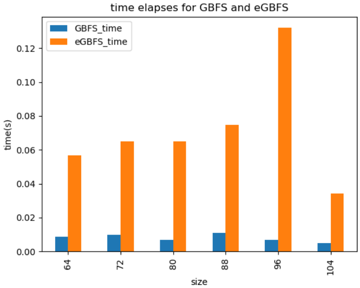

GBFS
eGBFS
Time Complexity/Nodes Visited
300
200
100
10
20
30
40
50
IDS
250000
200000
150000
100000
50000
0
Size


GBFS
eGBFS
Time Complexity/Nodes Visited
700
600
500
400
300
65
70
75
80
85
90
95
100
105
Size

Time Elapsed


time elapses for GBFS and eGBFS


time elapses IDS


time elapses for GBFS and eGBFS

The End