MAZE SOLVING USING BFS, GBFS, AND A* SEARCH

INTRODUCTION

This report presents the implementation and results of solving single-prize mazes using Breadth-First Search (BFS), Greedy Best-First Search (GBFS), and A* Search (A*). Each algorithm explores the maze differently to reach the prize efficiently.

IMPLEMENTATION DETAILS

The functions `single_bfs`, `single_gbfs`, and `single_astar` take a maze file as input and perform searches using different strategies. BFS ensures the shortest path, while GBFS and A* use heuristics to improve efficiency.

EXECUTION PROCESS

To run the search algorithms, execute the following command: python main.py

This script will solve three maze files and print the results

RESULTS

Below are the results of solving three different mazes using BFS.

```
#%
 #%
 #%
     용
ջ
 ##%
**Path Cost: 45, Nodes Expanded: 513**
### BFS Solution for 1prize-medium.txt:
8 8 8
      8##########
8 8 8 8 8 8 8 #####8 8 8 8###8 8 8
8######
  #####% % % ####### % % % % % %
%###% %
8#8 8 8 8 8 8 8
      8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8
%P## % %
     용용
**Path Cost: 94, Nodes Expanded: 611**
### BFS Solution for 1prize-large.txt:
$#$$$#$ $$$#$#$ $ $ $$$$ $ $ $ $ $ $ $$$ $$$ $$
왕 왕 왕
  응 응 응 응
  8 8 8 8 8 8
8 8 88 88 888 8 888 8
```

 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 8
 <td

```
% % % % % % % % % % % % % % ####% % %
8 8 8 8 8 ###8 8 8 8
8 8 8 8#88888 8
9 9
 8 8 8 8
```

Path Cost: 148, Nodes Expanded: 1259

Below are the results of solving three different mazes using GBFS.

```
### GBFS Solution for 1prize-open.txt:
9
           %P
                  용
용
           응#
                  용
용
           응#
                  용
응
           응#
           8######
응
읒
           응응응응응#
응
              응#
                  용
              응#
응
                  응
              응#
응
              응#
   #############
             응#
   #%%%%%%%%%%%#########
응
응
   #%
용
   #%
응
    #%
ջ
    #%
                  용
   #%
   ##%
**Path Cost: 45, Nodes Expanded: 149**
### GBFS Solution for 1prize-medium.txt:
8###### 8 8 8 8
                  #### %###########%
8#8 ##8 8 8 8 #### 8#####8 8 8 8###8 8 8
```

\$#\$ \$\$\$#\$\$\$ \$\$\$ \$\$\$ \$#\$\$\$#\$\$\$\$#\$\$ \$ \$\$\$ \$ \$\$\$ \$ \$\$\$ \$\$ 8 #####8 ### 8 8 8 8 8 8 8 8 \$#\$ \$ \$ \$####### \$ \$ \$ \$ \$ \$ \$ \$ 용 8# 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8#8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 \$#\$ \$ \$ \$ \$ \$ \$ \$ 8 8 8 8 % % % % % % % % % % % %P## % % % % % %

Path Cost: 114, Nodes Expanded: 134

GBFS Solution for 1prize-large.txt:

8 8 888 8 8 8 8 8 ####8 8 8 8 8###8 8 8 8 8 8 #8 8 8 8 8 8 8 8 8 8 8 % % %% %% %%% % %%% % 8 8 8 8 8 8 8 8 9 9 9 9 9 8 8 88888 8 8#8 888 8 8 888 888 8 88 \$ \$\$\$\$\$ \$\$\$ \$\$\$ \$\$\$ \$ \$\$\$\$\$ \$\$\$ \$\$\$ \$ 8 8 8 8#8 응 응 응 8 8 8 8 응응 응응응 응 유 \$#\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ 응 8 8 8 8 응#응 8 8 8 8 8 8 8 8 8 8 8 8 # 8#8 8 8 %%% % %%% % %%% %%% 응응응응 응#응응응응 % % %%% % % %%% % %%%%%%% %%%#%%%#% % % % \$###\$###\$###\$ \$ \$ \$ \$ 8 8 8 8 #8###8 8 8 8 #%###%###% % % % % 응 응 ## %#####\$#######\$#%#% %###% %###%#% 응 응응응응 % % %%\$#\$%\$\$#\$%\$\$#\$% %%\$%\$%\$%\$% %%\$%\$%\$#\$%\$ % %%\$ % % %% \$#\$#\$#\$%\$% %%\$#\$%\$%\$#\$ % ##### %############## % % 응 응 8###8###########

```
**Path Cost: 222, Nodes Expanded: 278**
```

Below are the results of solving three different mazes using A* Search.

```
### A* Search Solution for 1prize-open.txt:
%P
          응#
용
          응#
          응#
          8######
양
          응응응응응용#
             응#
엉
             응#
                 용
   ########### %#
   #%%%%%%%%%%%%##########
용
   #%
용
   #%
   #%
   #%
   #%
   ##%
**Path Cost: 45, Nodes Expanded: 238**
### A* Search Solution for 1prize-medium.txt:
```

엉	00	િ	응우	5	00	ુ	응	엉	9	응용	58	용역	o o	응원	능용	용	olo Olo	응용	응용		응	엉	엉	૭ જ	s#	બ	응‡	‡ %	용	응#	응유	5	응응	j	응원	5 %	#	응#	‡ %	olo	용
િ	90					્રા		엉				Ş	olo					앙			용		#	##	##		응‡	##	#	##	응		90	í	엉	양	#	##	‡ %	96	: %
양	양		응우	응	:	olo	응우	응용	ુ	9	5 S	Ş	응	응용	능용	용의	응용	양	양	용	응원	5 S	#	응용	응용		용	양	용	0/0	બ	엉	90	5	બ	양	양	응원	58	96	: %
00	90			90			엉					ç	olo O					엉			##	#	#	90			િ			90	양	용	90	5				엉		96	· %
90	90		응유	응응		응응	90	%	용의	00	양	용우	20	9	90	용의	20	응원	응응	િ	# 9	8		90	90	엉	%	90		응용	ુ	90	olo	용	િ	90	양	90	89	ું હ	용
90			્રે				90				양			9	90	# :	##	##	##	#	#	જ		90	90			90				90			િ	90			90	olo	용
양	양	양	용	9	용	ુ	용위	응용!	용의	olo	용	응유	응응	양	양	# 5	응응	양	양	양	용	양	양	응용	응		응원	등 응	용	응응	, ,	응	응응	í	응원	58		응원	58	90	응
00				90	#	##	##	##	# 5	olo Olo			#	##	##	# 5	olo O	엉	90								િ			90			90	5				엉		96	· %
양	양	용	응우	응	#	응응	응우	응응:	# 5	응용	5 S	용우	è#	응용	능용	9	olo	양	양		응원	5 S	ુ	응용	응응	ુ	용	양		0/0	용유	응응	90	; 응	બ	양	양	엉	용위	응용	: %
양	양			9	#	olo		왕:	# =	##	#	# ‡	##		양	9	olo		양		용				양			양		0/0	બ								양		ક
엉	90		응유	응	#	ુ	89	응응	9	કુ ક	5 S	용역	20	બ	olo	9	ટું ક	응원	응응		90	엉		응용	응응		90	90		응응	ે	90	응응	: %	િ	olo	엉	% જ	58	00	응응
00			응‡	##	#	9	엉					ç	olo O	엉	엉			엉			િ	양			용		િ	엉				용				엉			90	96	· %
양	양	용	응‡	‡ %	용	olo	양	엉	9	응용	5 S	Ş	응	િ	양	용의	olo	양	양	용	용	જ	ુ	ુ	엉		용	양	용	응응	응유	응응	응응	í	응원	58		엉	양	96	: %
00	90	#	##	‡%			엉	엉			양			엉	엉			엉	90					90			િ			90			90	5	엉			엉		96	· %
응‡	##	#	응유	응응		96	90	엉	9	00	양	ç	20	9	90	9	응응	엉	엉		응용	8	엉	응용	응응		응원	응		00	% જ	응응	olo	5	િ	90	양	% જ	응용	ું હ	용
응‡	‡%			00		ુ		엉	9	00		Ş	20		olo	9	200													00	બ		olo	5	િ	olo					용
응‡	‡%	િ	엉	00	용	ુ	89	응응	9	응용	5 S	ç	20	응용	능용	9	200	용원	응응		િ	90	엉	응용	응응	બ	q	응	응:	00	બ	엉	응응	ક	િ	90		olo	89	ું લ	용
응‡	##	#	્રે						9	00		ç	20	9		9	20	엉			%							90			બ	90				90		90	90		િ
용원	5 응	#	응유	을 응		90	응우	응용	용의	olo	양	응유	응응	%	양	용의	응응	양	용	용	용	양	양	응용	응용	양	용	용		응응	용	양	90	응	응원	58	양	엉	양	90	응응
%I	2#	#		00					9	90									양								િ				엉								90		િ
% જ	5 응	용	응우	3 %	용	응응	응유	응용!	응의	응용	58	용위	응	99	;응	용의	응용	용원	응용	용	응원	58	양	응원	응용	િ	응원	응	용	응용	:89	응용	응응	응	응용	58	양	응원	: 응 ?	응용	응응

Path Cost: 94, Nodes Expanded: 335

A* Search Solution for 1prize-large.txt:

. . .

 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888
 8888

응 응 응 응 응 응 88 88 888 8 888 8 8######## % % 응 응 응#응 응 8 8 8 #8 8 8 #### 8 8 8 ######## # ## % %%% % %%% %%%%% %%%%%%% %%% %####%% %%%#%%%%%#% %%% % % 8 8 8888#8888 888 8 응 응 8###### 8 8 용 용 용 용 88888 8 88888 % % %%% % % %%% % %%%%%%% %%% %%%#% % % % % 용 응 응 응 응 응 응 용 용 8 8 8 8 8 8 응 응 용 8###8

Path Cost: 148, Nodes Expanded: 1113

CONCLUSION

Breadth-First Search (BFS), Greedy Best-First Search (GBFS), and A* Search successfully find the prize in the maze. BFS guarantees the shortest path, while GBFS and A* use heuristics to improve efficiency. The results show the path cost and number of nodes expanded, which are important performance metrics for search algorithms.