

Part 1: Basic Path Finding

Group 1:

Breadth-first Search:

```
PS C:\Users\kk2\Desktop\maze> python main.py bfs smallMaze.lay
-Original Maze-
%%%%%%%%%
%      %      %      %.%
%% %  %% %  %% %  %
% % %      %  % %  %
% % % % %% % %% % %%
%      %  %      % %
% %% % % % % % %% %
% % %  % % %      % %
% % % %% % %% % % %%
%P %      % %      %
%%%%%%%%%

-Solution Maze-
%%%%%%%%%
%      %      %  ...%.%
%% %  %% %  %% %.%.%
% % %.....%  %.%...%
% % %.% %%.% %.% %
%. ....%  %..... % %
%. % % % % % % % %
%. % %  % % %      % %
%. % % %% % %% % % %%
%P %      % %      %
%%%%%%%%%

Solution Cost 37
Max Tree Depth 112
Max Frontier Size 6
Nodes Expanded 114
```

```
PS C:\Users\kk2\Desktop\maze> python main.py bfs mediumMaze.lay
```

-Original Maze- -Solution Maze-

[illegible]

Nodes Expanded 344

-Solution Maze-

[illegible]

-Original Maze-

```
Solution Cost 327
Max Tree Depth 1116
Max Frontier Size 10
Nodes Expanded 1125
```

[illegible]

```
PS C:\Users\kk2\Desktop\maze> python main.py bfs openMaze.lay
```

-Original Maze-

[illegible]

Solution Cost 75

Max Tree Depth 574

Max Frontier Size 14

Nodes Expanded 575

-Solution Maze-

[illegible]

Group 2:

A* Search

```
PS C:\Users\kk2\Desktop\maze> python main.py ast smallMaze.lay
```

-Original Maze-

[illegible]

Solution Cost 37

Max Tree Depth 99

Max Frontier Size 8

Nodes Expanded 98

-Solution Maze-

[illegible]

PC 914 111018 11 1

-Solution Maze-

[illegible]

Nodes Expanded 198

-Original Maze-

```
Solution Cost 327
Max Tree Depth 927
Max Frontier Size 12
Nodes Expanded 503
```

[illegible]

-Solution Maze-

[illegible]

Solution Cost 75
Max Tree Depth 440
Max Frontier Size 28
Nodes Expanded 439

Part 2: Search with different cost functions

1: $c_1 = \frac{1}{2}x$

```
PS C:\Users\kk2\Desktop\maze> python main.py uc mediumMaze.lay
```

Please enter a valid cost function (0,1 or 2):

-->1

-Original Maze-

[illegible]

-Solution Maze-

[illegible]

Solution Cost 129

Max Tree Depth 372

Max Frontier Size 11

Nodes Expanded 379

2: $c_2 = 2^x$

```
PS C:\Users\kk2\Desktop\maze> python main.py uc mediumMaze.lay
Please enter a valid cost function (0,1 or 2):
```

-->2

-Original Maze-

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	8												

Solution Cost 73

Max Tree Depth 127

Max Frontier Size 17

Nodes Expanded 141

-Solution Maze-

[illegible]

Part 3: Search with multiple goals

Breadth-first search

```
PS C:\Users\kk2\Desktop\maze> python main.py bfs tinySearch.lay
```

-Original Maze-

-Solution Maze-

~~0/0/0/0/0/0/0/0/0~~
~~/0/0/0/0/0/0/0/0/0~~

0/0/0/0/0/0/0/0/0/
0/0/0/0/0/0/0/0/0

% % %

% . . . % . . . %

0% 0% 0%0%0% 0%

0% 0% 0%0% 0%
/0 /0. /0/0/0. /0

% % . . %

% % %

% %P%

% %P%.%

% . . . %

% %

% % %/%/%/%/%
/% /% %/%/%/%/%

0% 0% 0%0%0%0%
% % %0%0%0%0%

% . . . %

% %

0/0/0/0/0/0/0/0/0/
/0/0/0/0/0/0/0/0/00/0/0/0/0/0/0/0/0/
/0/0/0/0/0/0/0/0/0/

Solution Cost 43

Max Tree Depth 74

Max Frontier Size 4

Nodes Expanded 97

[illegible]

-Solution Maze-

[illegible]

```
PS C:\Users\kk2\Desktop\maze> python main.py bfs trickySearch.lay  
-Original Maze-  
%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/  
%.          ..%   %  
%.%/%.%/%.%/%.%/%.%/%.%/%.%/%.%/%.  
%           P     % %  
%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/  
%.....      %  
%/%/%/%/%/%/%/%/%/%/%/%/%/%/%/  
Solution Cost 69  
Max Tree Depth 121  
Max Frontier Size 5  
Nodes Expanded 140
```

-Solution Maze-

```

%.....%.....%
%.....%.....%
%.....P.....%
%.....%
%.....%

```

```
PS C:\Users\kk2\Desktop\maze> python main.py ast tinySearch.lay  
-Original Maze-                               -Solution Maze-  
%%%%%%%%%          %%%%%%%%%  
%. %. %           %...%...%  
% % %%% %        % %.%%%.%  
% % . .%         % %.....%  
% %%P%% %        % %%P%%.%  
%. .. .%         %.....%  
% % %%%%         % %.%%%%  
%. . .%         %.....%  
%%%%%%%%%          %%%%%%%%%  
  
Solution Cost 57  
Max Tree Depth 89  
Max Frontier Size 7  
Nodes Expanded 6
```

-Solution Maze-

-Solution Maze-

[illegible]

A* search with Euclidean Distance Heuristic

Used Euclidean distance instead of Manhattan distance and got following result

[illegible]

Which yield almost same result as Manhattan distance on big maze, just Max Tree Depth increased by '37' and Nodes Expanded increased by '1' on big maze compared to Manhattan distance. Since Euclidean distance is the shortest path between source and destination which is a straight line, it doesn't perform better than Manhattan distance in this Pac-man problem i.e. we can't go diagonally in the maze (which is the case for Euclidean distance).

A* search with CHEBYSHEV Distance Heuristic

Used Chebyshev distance instead of Manhattan distance and got the following results for big maze

[illegible]

Which yield almost same result as Manhattan distance on big maze, just Max Tree Depth increased by '46' and Nodes Expanded increased by '3' on big maze compared to Manhattan distance.

Therefore Manhattan distance performance better than other distance functions in case of Pac-man.