CS 440 Assignment 1

R section 3 credits

Group members:
Chendi Lin, Yulun Yao, Zhuoyue Wang

1. Overview

In this assignment, we design and use different searching algorithms to solve "food pellets" puzzles.

In the first part, we need to find the shortest path to the puzzle has only one goal state by a given start state. To approach this problem, we use BFS, DFS, Greedy and A* search algorithms to find solution paths and their path cost and number of expanded nodes.

In the second part, we are asked to solve the maze with multiple goals. As a starting point, we apply BFS on “tinySearch” to double check that our state representation. Then, we use graph as the maze representation to compress the number of states, and use the total length of the Minimum Spanning Tree of the graph as the heuristic to improve the searching efficiency. The result turns out to be pretty good.

For the extra credit part, we utilize the similar strategy, but with a different way to record the dots collection and use an overestimate heuristic to decrease the computational expense but as a trade off, only suboptimal solution can be found.

2. Work distribution

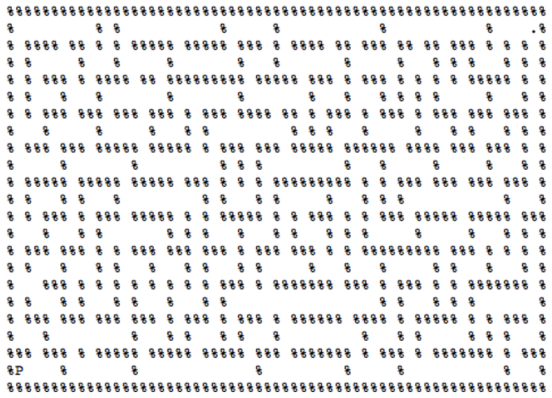
Chendi Lin: Greedy best-first search algorithm & A* search algorithm, 1.2 multiple dot search, extra credit, report

Yulun Yao: A* search algorithm, 1.2 multiple dot search, report

Zhuoyue Wang: Breadth-first search & Depth-first search, read_maze, report

3. Section 1.1 Basic pathfinding

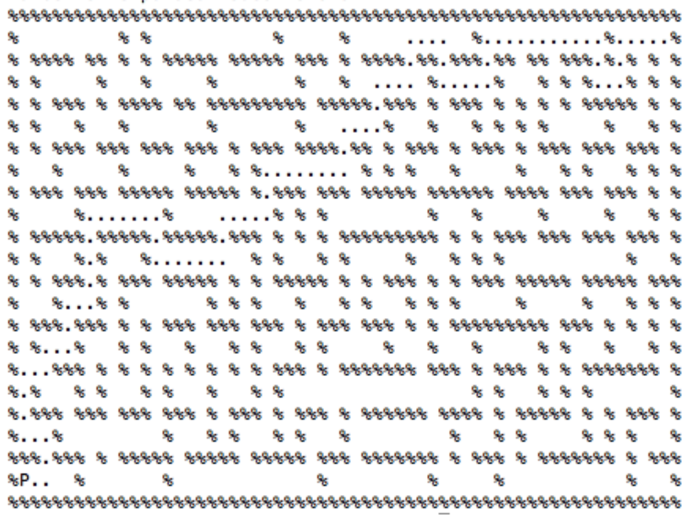
<u>Input: mediumMaze</u>



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         % %           %     %           %           %    .%
% %%%% %% % % %%%%% %%%%% %%% % %%%% %% %%% %% %% %%% % % % %
% %     %   %     %       %   %       %     %   % % %   % % %
% % %%% % %%%% %% %%%%%%%%% %%%%% %%% % %%% % % % % %%%%% % %
% %   %   %       %       %       %   %   % % % %     %   % %
% % %%% %%% %%% %%% % %%% %%%% %% % %%% % %%% % %%% %%% %%% %
%   %     %     %   % %         % % %   %     %   % %   % % %
% %%% %%% %%%%% %%%%% % %%% %%% %%%%% %%%%%% %%%% %%% %%% % %
%     %       %         % % %         %   %     %     %   % %
% %%%%% %%%%% %%%%% %%% % % % %%%%%%%%% % % %%% %%% %%% %%% %
% %   % %   %         % %   % %     %   % % %           %   %
% % %%% % %%% %%%%% % % %%%%% % % %%% % % %%% %%%%% %%%%% %%%
%   %   % %       % % %   %   % %   % % %     %     %   % % %
% %%% %%% % % %%% %%% %%% % %%% %%% % % %%%%%%%%% %%% % % % %
% %   %   % %   %   % %   % %     %   %   %     % %   %   % %
%   %%% % % % % % % % % %%% % %%%%%%% %%% % %%% % % %%%%%%% %
% %   % %   % %   %   % %                 % %   % % %       %
% %%% %%% %%% %%% % %%% % %%% % %%%%%% %%%% % %%%%% % % %%% %
%   %         %   % %   % %   %         %   % %     % % %   %
%%% %%% % %%%%% %%%%% %%%%% %%% %%%%%%% % %%% % %%%%%%% % %%%
%P    %       %             %         %     %           %   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Breadth-first search:

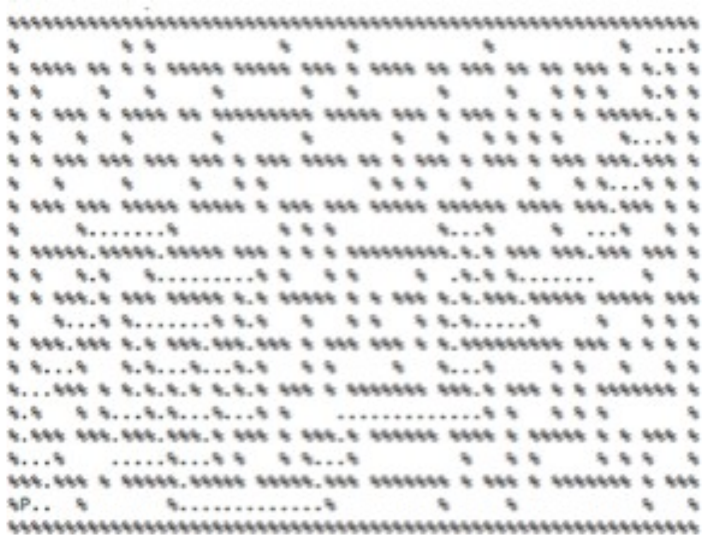
Solution:
Path Cost: 94
Number of nodes expanded: 610



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         % %           %     %     ....  %...........%.....%
% %%%% %% % % %%%%% %%%%% %%% % %%%%.%%.%%%.%% %% %%%.%.% % %
% %     %   %     %       %   %  .... %.....%   % % %...% % %
% % %%% % %%%% %% %%%%%%%%% %%%%%.%%% % %%% % % % % %%%%% % %
% %   %   %       %       %   ....%   %   % % % %     %   % %
% % %%% %%% %%% %%% % %%% %%%%.%% % %%% % %%% % %%% %%% %%% %
%   %     %     %   % %........ % % %   %     %   % %   % % %
% %%% %%% %%%%% %%%%% %.%%% %%% %%%%% %%%%%% %%%% %%% %%% % %
%     %.......%    .....% % %         %   %     %     %   % %
% %%%%%.%%%%%.%%%%%.%%% % % % %%%%%%%%% % % %%% %%% %%% %%% %
% %   %.%   %.......  % %   % %     %   % % %           %   %
% % %%%.% %%% %%%%% % % %%%%% % % %%% % % %%% %%%%% %%%%% %%%
%   %...% %       % % %   %   % %   % % %     %     %   % % %
% %%%.%%% % % %%% %%% %%% % %%% %%% % % %%%%%%%%% %%% % % % %
% %...%   % %   %   % %   % %     %   %   %     % %   %   % %
%...%%% % % % % % % % % %%% % %%%%%%% %%% % %%% % % %%%%%%% %
%.%   % %   % %   %   % %                 % %   % % %       %
%.%%% %%% %%% %%% % %%% % %%% % %%%%%% %%%% % %%%%% % % %%% %
%...%         %   % %   % %   %         %   % %     % % %   %
%%%.%%% % %%%%% %%%%% %%%%% %%% %%%%%%% % %%% % %%%%%%% % %%%
%P..  %       %             %         %     %           %   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

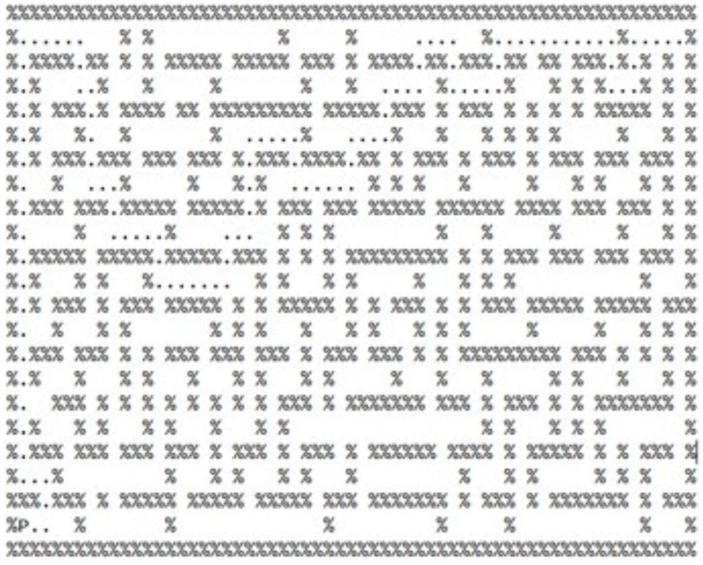
Depth-first search:

Solution:
Path Cost: 162
Number of nodes expanded: 212



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         % %           %     %           %           %  ...%
% %%%% %% % % %%%%% %%%%% %%% % %%%% %% %%% %% %% %%% % %.% %
% %     %   %     %       %   %       %     %   % % %   %.% %
% % %%% % %%%% %% %%%%%%%%% %%%%% %%% % %%% % % % % %%%%%.% %
% %   %   %       %       %       %   %   % % % %     %...% %
% % %%% %%% %%% %%% % %%% %%%% %% % %%% % %%% % %%% %%%.%%% %
%   %     %     %   % %         % % %   %     %   % %...% % %
% %%% %%% %%%%% %%%%% % %%% %%% %%%%% %%%%%% %%%% %%%.%%% % %
%     %.......%         % % %         %...%     %  ...%   % %
% %%%%%.%%%%%.%%%%% %%% % % % %%%%%%%%%.%.% %%% %%%.%%% %%% %
% %   %.%   %.........% %   % %     %  .%.% %.......    %   %
% % %%%.% %%% %%%%% %.% %%%%% % % %%% %.%.%%%.%%%%% %%%%% %%%
%   %...% %.......% %.%   %   % %   % %.%.....%     %   % % %
% %%%.%%% %.% %%%.%%%.%%% % %%% %%% % %.%%%%%%%%% %%% % % % %
% %...%   %.%...%...%.%   % %     %   %...%     % %   %   % %
%...%%% % %.%.%.% %.%.% %%% % %%%%%%% %%%.% %%% % % %%%%%%% %
%.%   % %...%.%...%...% %    .............% %   % % %       %
%.%%% %%%.%%%.%%%.% %%% % %%%.% %%%%%% %%%% % %%%%% % % %%% %
%...%    .....%...% %   % %...%         %   % %     % % %   %
%%%.%%% % %%%%%.%%%%% %%%%%.%%% %%%%%%% % %%% % %%%%%%% % %%%
%P..  %       %.............%         %     %           %   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Greedy best-first search:
Solution:
Path Cost: 114
Number of nodes expanded: 140



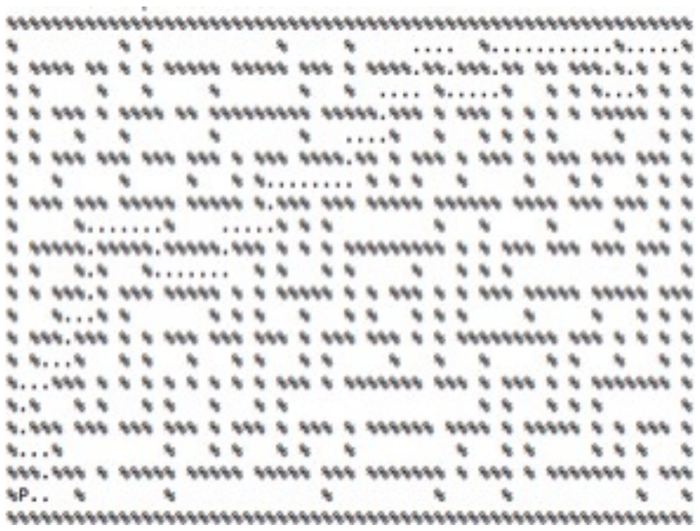
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%......   % %           %     %     ....  %...........%.....%
%.%%%%.%% % % %%%%% %%%%% %%% % %%%%.%%.%%%.%% %% %%%.%.% % %
%.%   ..%   %     %       %   %  .... %.....%   % % %...% % %
%.% %%%.% %%%% %% %%%%%%%%% %%%%%.%%% % %%% % % % % %%%%% % %
%.%   %.  %       %  .....%   ....%   %   % % % %     %   % %
%.% %%%.%%% %%% %%% %.%%%.%%%%.%% % %%% % %%% % %%% %%% %%% %
%.  %  ...%     %   %.%  ...... % % %   %     %   % %   % % %
%.%%% %%%.%%%%% %%%%%.% %%% %%% %%%%% %%%%%% %%%% %%% %%% % %
%.    %  .....%    ...  % % %         %   %     %     %   % %
%.%%%%% %%%%%.%%%%%.%%% % % % %%%%%%%%% % % %%% %%% %%% %%% %
%.%   % %   %.......  % %   % %     %   % % %           %   %
%.% %%% % %%% %%%%% % % %%%%% % % %%% % % %%% %%%%% %%%%% %%%
%.  %   % %       % % %   %   % %   % % %     %     %   % % %
%.%%% %%% % % %%% %%% %%% % %%% %%% % % %%%%%%%%% %%% % % % %
%.%   %   % %   %   % %   % %     %   %   %     % %   %   % %
%.  %%% % % % % % % % % %%% % %%%%%%% %%% % %%% % % %%%%%%% %
%.%   % %   % %   %   % %                 % %   % % %       %
%.%%% %%% %%% %%% % %%% % %%% % %%%%%% %%%% % %%%%% % % %%% %
%...%         %   % %   % %   %         %   % %     % % %   %
%%%.%%% % %%%%% %%%%% %%%%% %%% %%%%%%% % %%% % %%%%%%% % %%%
%P..  %       %             %         %     %           %   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A* search:

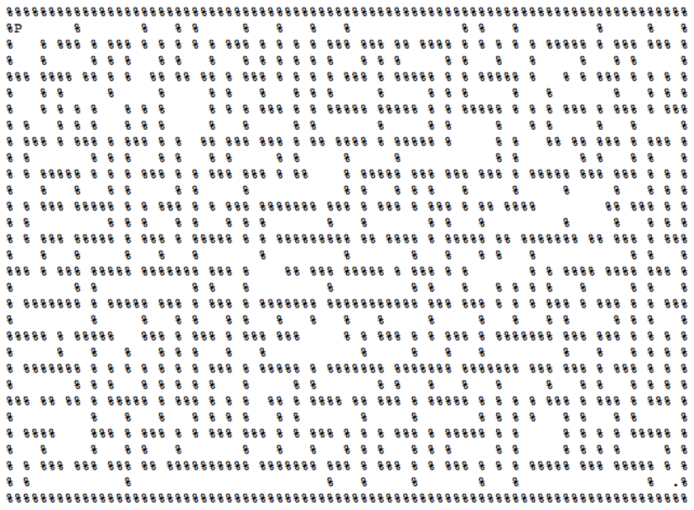
Solution:

Path Cost: 94

Number of nodes expanded: 339



Input: bigMaze

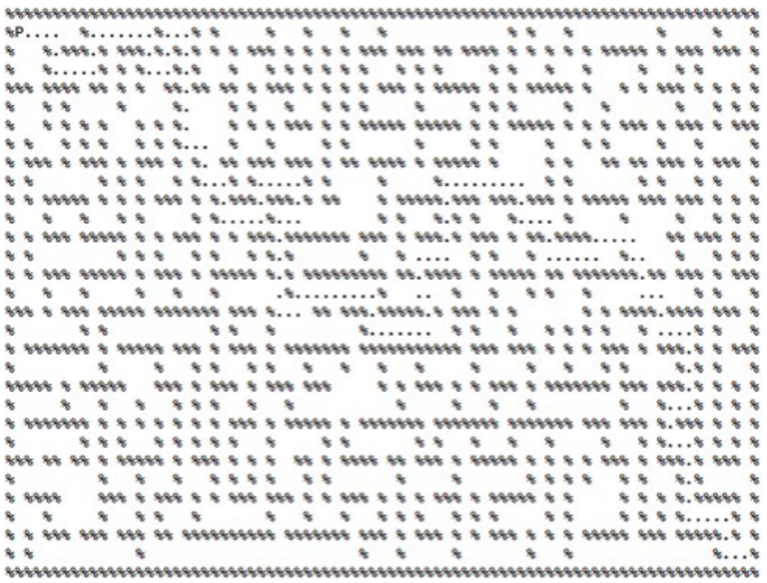


Breadth-first search:

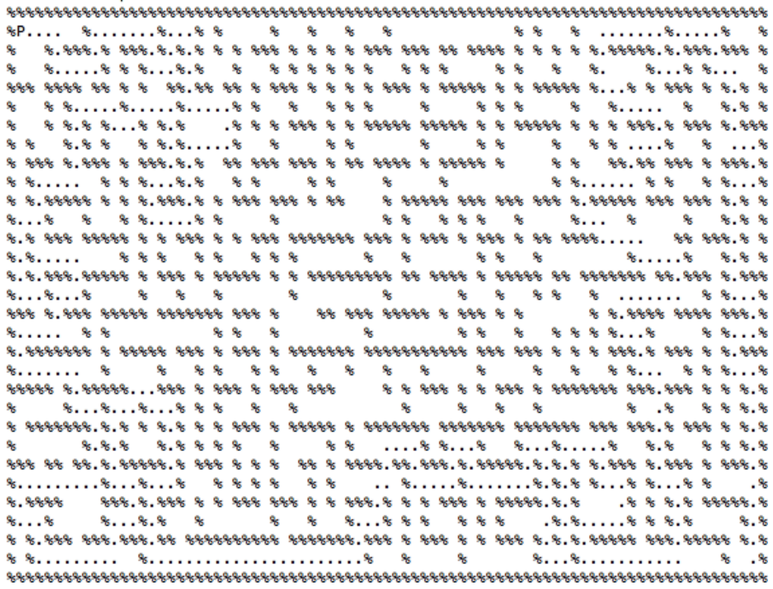
Solution:

Path Cost: 148

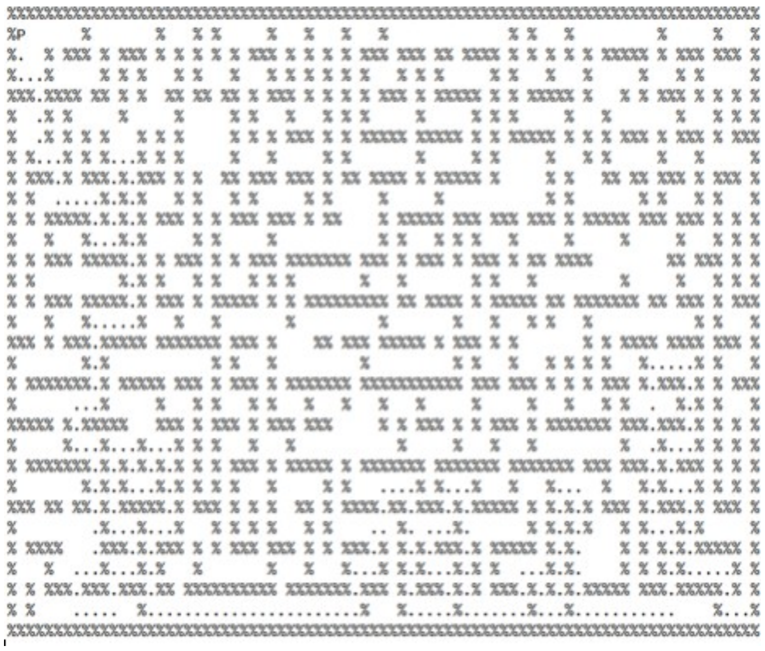
Number of nodes expanded: 1256



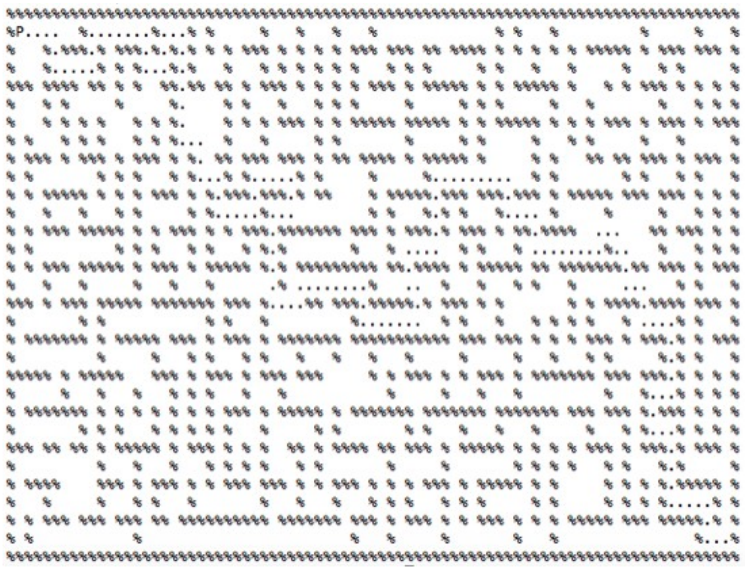
Depth-first search:
Solution:
Path Cost: 386
number of nodes expanded: 540



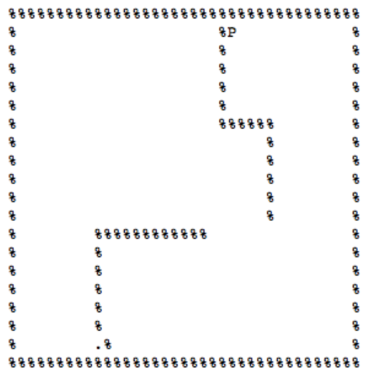
Greedy best-first search:
Solution:
Path Cost: 234
Number of nodes expanded: 297



A* search:
Solution:
Path Cost:148
Number of nodes expanded: 1148

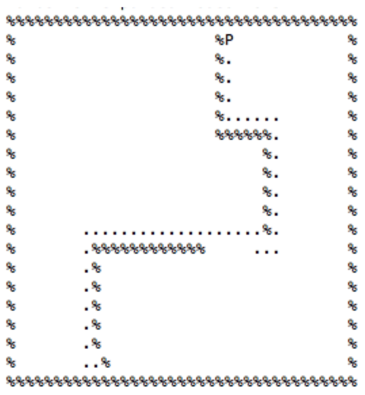


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%P....  %.......%...% %     %   %   %   %             % %   %         %     %   %
%   %.%%%.% %%%.%.%.% % % %%% % % % % %%% %%% %% %%%% % % % % % %%%%% % %%% %%% %
%   %.....% % %...%.%   %   % % % % % %   % % %     % %   %   %     %   % %     %
%%% %%%% %% % %  %%.%% %% % %%% % % % % %%% % %%%%% % % %%%%% %   % % %%% % % % %
%   % %     %     %.    % %   %   % % %     %     % % %     %   %       %   % % %
%   % % % %   % % %.    % % % %%% % % %%%%% %%%%% % % %%%%% % % % %%% % %%% % %%%
% %   % % %   % % %...  %   %     % %       %     % %     %   % %     %   %     %
% %%% % %%% % %%% % %. %% %%% %%% % %% %%%% % %%%%% %     % %   %% %% %%% % %%% %
% %       % % %   % %...% %.....% %     %     %.........  % %       % %   % %   %
% % %%%%% % % % %%% % %.%%%.%%%.% %%    % %%%%%.%%% %%%.%%% % %%%%% %%% %%% % % %
%   %   %   % %     % %.....%...        % %   %.% %   %.... %     %     %   % % %
% % %%% %%%%% % % %%% % % %%%.%%%%%%% %%% % %%%.% %%% % %%.%%%%  ...   %% %%% % %
% %         % % %   % %   % %.%       %   % ....  % %   % ........%..   %   % % %
% % %%% %%%%% % %%% % %%%%% %.% %%%%%%%%% %%.%%%% % %%%%% %% %%%%%%.%%  %%% % %%%
%   %   %     %   %   %      .% ........%   ..  %   %   % %   %     ...   % %   %
%%% % %%% %%%%% %%%%%%% %%% %....%% %%%.%%%%%.% %%% % %       % % %%%%.%%%% %%% %
%       % %           % %   %         %.......  % %   %   % % % %   % ....% %   %
% %%%%%%% % %%%%% %%% % %%% % %%%%%%% %%%%%%%%%%% %%% %%% % % % %%% % %%%.% % %%%
%         %     %   % %   % %   %   %   %   %     %     %   %   % %     %.% %   %
%%%%% % %%%%%   %%% % %%% % %%% %%%     % % %%% % % %%% % %%%%%%% %%% %%%.% % % %
%     %   %   %   % % %   %   %           %     %   %   %         %   %...% % % %
% %%%%%%% % % % % % % % %%% % %%%%% % %%%%%%% %%%%%%% %%%%%%% %%% %%% %.%%% % % %
%       % % %   % % % % %   %     % %       % %   %   %   %     %   % %...% % % %
%%% %% %% % %%%%% % %%% % % %  %% % %%%% %% %%% % %%%%% % % % % %%% % %%%.% %%% %
%         %   %   %   % % % %   % %       %     %       % % % %   % %   %.%     %
% %%%%    %%% % %%% % % %%% %%% % % %%% % % % %%% % %%%%% % %     % % % %.%%%%% %
%   %     %   % %   %       %   %   %   % % %   % % %     % %     % % % %.....% %
% % %%% %%% %%% %% %%%%%%%%%% %%%%%%% %%% % %%% % % %%% % % % %%%%% %%% %%%%%.% %
% %           %                       %   %     %       %   %               %...%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



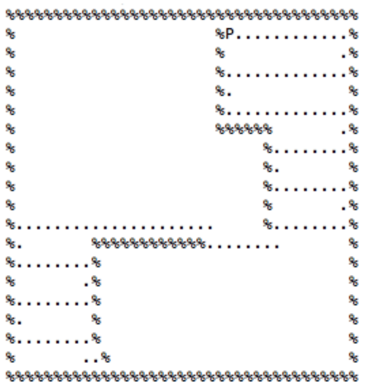
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P            %
%                     %             %
%                     %             %
%                     %             %
%                     %             %
%                     %%%%%%        %
%                          %        %
%                          %        %
%                          %        %
%                          %        %
%                          %        %
%        %%%%%%%%%%%%               %
%        %                          %
%        %                          %
%        %                          %
%        %                          %
%        %                          %
%        .%                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Input: openMaze



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P            %
%                     %.            %
%                     %.            %
%                     %.            %
%                     %......       %
%                     %%%%%%.       %
%                          %.       %
%                          %.       %
%                          %.       %
%                          %.       %
%       ...................%.       %
%       .%%%%%%%%%%%%     ...       %
%       .%                          %
%       .%                          %
%       .%                          %
%       .%                          %
%       .%                          %
%       ..%                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Breadth-first search:
Solution:
Path Cost: 45
Number of nodes expanded: 524



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P............%
%                     %            .%
%                     %.............%
%                     %.            %
%                     %.............%
%                     %%%%%%       .%
%                          %........%
%                          %.       %
%                          %........%
%                          %       .%
%.....................     %........%
%.       %%%%%%%%%%%%........       %
%........%                          %
%       .%                          %
%........%                          %
%.       %                          %
%........%                          %
%       ..%                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Depth-first search:
Solution:
Path Cost: 125
Number of nodes expanded: 335

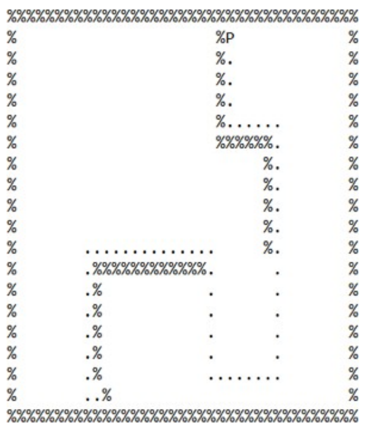
Greedy best-first search:

Solution:

Path Cost: 55

Number of nodes expanded: 231

Note: It is clear here that Greedy does not give
a optimal solution, which is one of the disadvantages
of greedy best-first search



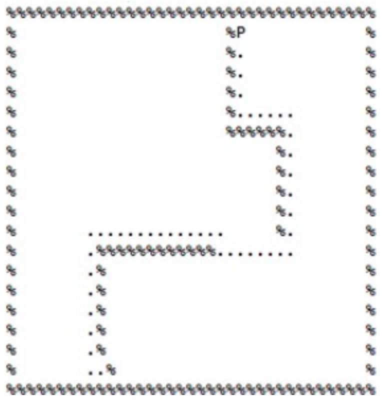
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P            %
%                     %.            %
%                     %.            %
%                     %.            %
%                     %......       %
%                     %%%%%%.       %
%                          %.       %
%                          %.       %
%                          %.       %
%                          %.       %
%       ..............     %.       %
%       .%%%%%%%%%%%%.      .       %
%       .%           .      .       %
%       .%           .      .       %
%       .%           .      .       %
%       .%           .      .       %
%       .%           ........       %
%       ..%                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A* search:

Solution:

Path Cost: 45

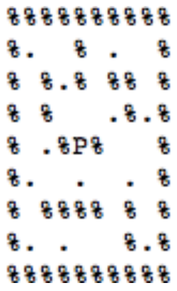
Number of nodes expanded: 238



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P            %
%                     %.            %
%                     %.            %
%                     %.            %
%                     %......       %
%                     %%%%%%.       %
%                          %.       %
%                          %.       %
%                          %.       %
%                          %.       %
%       ..............     %.       %
%       .%%%%%%%%%%%%........       %
%       .%                          %
%       .%                          %
%       .%                          %
%       .%                          %
%       .%                          %
%       ..%                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

4. Section 1.2 Search with multiple dots

Breadth-first search for tinySearch



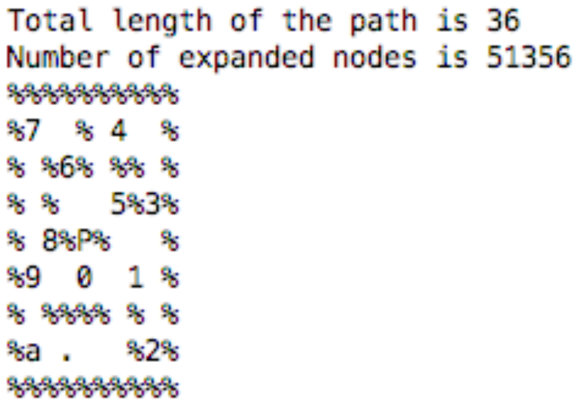
```
%%%%%%%%%%
%.  % .  %
% %.% %% %
% %   .%.%
% .%P%   %
%.  .  . %
% %%%% % %
%. .   %.%
%%%%%%%%%%
```

Breadth-first search:

Solution:

Path Cost: 36

Number of nodes expanded: 51356



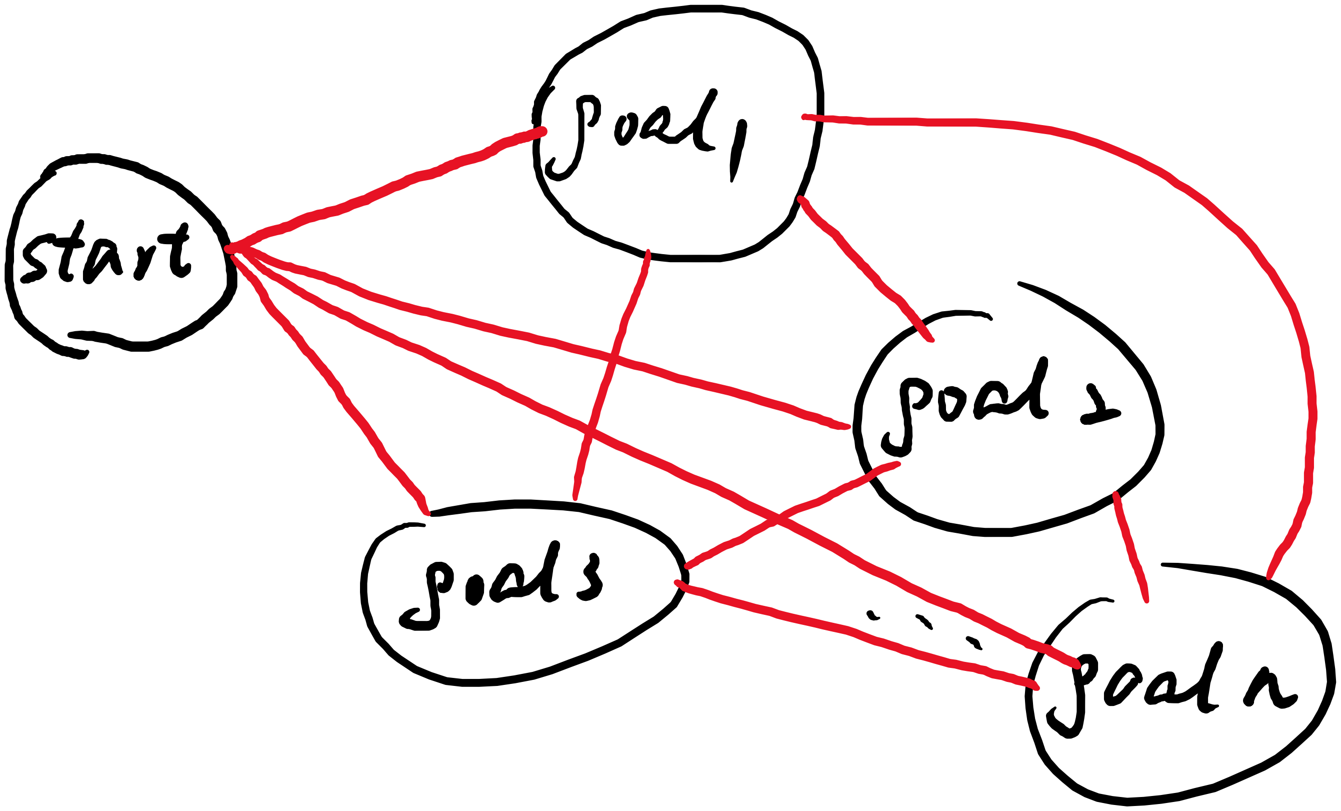
```
Total length of the path is 36
Number of expanded nodes is 51356
%%%%%%%%%%
%7  % 4  %
% %6% %% %
% %   5%3%
% 8%P%   %
%9  0  1 %
% %%%% % %
%a .   %2%
%%%%%%%%%%
```

A* Search:

Heuristic Algorithm Description:

For this part, we made some changes to improve the efficiency. We firstly set up a graph, with starting point and all goals as the nodes in the graph, and the shortest paths length between them as the edges of the graph. The time complexity here is O(n^2) and it is super fast even with "bigDots". Then instead of having the whole maze as our searching objective, we only search the graph, so the number of all the states becomes way smaller than before.

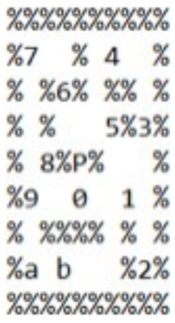
Now the problem becomes



with each black circle as a node, and each red line as a path in the problem. In this way, the needed memory can also be much smaller. Before to check repeated states, the explored set is initialized as a 3D array, with each dimension meaning x, y, and the collected dots. The collected dots are represented by a string composed of '0' and '1'. When goal i is collected, the ith char will be modified from '0' to '1'. Then by converting the string, which can also be regarded as a binary number, to an integer, we can find the location where this state should be recorded. But it requires huge amount of memories. Say the maze is m*n, and the number of dots is k. We need a m*n*2^k memory as the explored set. However, by converting this problem to a TSP, we do not need to track its x and y coordinate anymore. We only need to know which goal we are at right now. So now the space complexity becomes k*2^k. Because for “tinySearch”, “smallSearch”, and “mediumSearch”, k is much smaller than m and n, it becomes much more efficient.

The heuristic here is the total length of the Minimum Spanning Tree (MST) of the graph. It is an admissible heuristic because essentially, after setting up the problem as a graph, we are solving a Travelling Salesman Problem (TSP), where the solution is the shortest path with which the salesman only passes each node once. So it is clear that the solutions of TSP is the subset of the solutions of MST, because in MSP we only need to connect all the nodes. Thus, the actual heuristic, which is the result of TSP, is always larger than the estimated heuristic, which is the result of MST. This heuristic is also consistent, because obviously, whenever we expand a new node, a goal is collected, and the total length of MST will be shorter. With a admissible and consistent heuristic, the results are optimal.

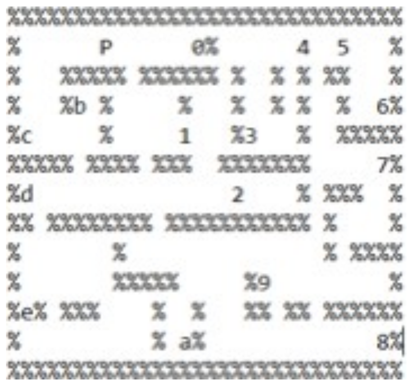
Input: tinySearch



```
%%%%%%%%%%
%7  % 4  %
% %6% %% %
% %   5%3%
% 8%P%   %
%9  0  1 %
% %%%% % %
%a b   %2%
%%%%%%%%%%
```

Solution:
Path
Cost:36
Number of nodes expanded: 54

Input: smallSearch

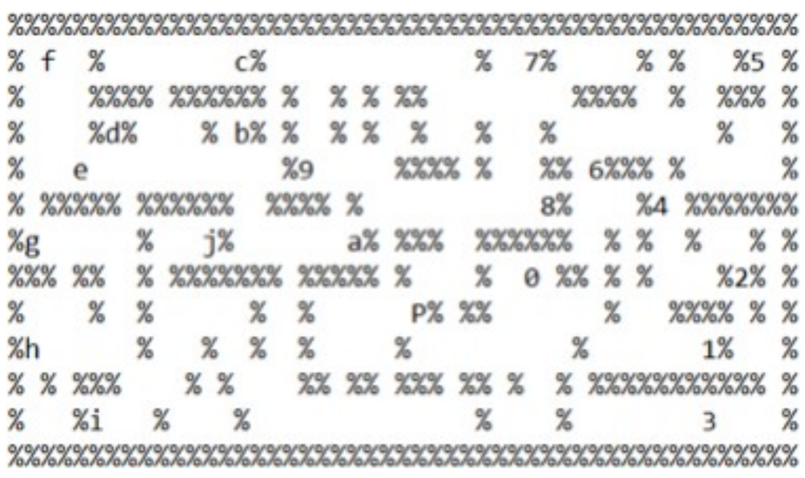


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      P      0%      4  5   %
%   %%%%% %%%%%% %  % % %%   %
%   %b %     %   %  % %  % 6%
%c     %     1   %3   %  %%%%%
%%%%% %%%% %%%  %%%%%%%     7%
%d               2    % %%%  %
%% %%%%%%%% %%%%%%%%%%% %    %
%       %               % %%%%
%       %%%%%    %9          %
%e% %%%    %  %  %% %%  %%%%%%
%          % a%             8%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Solution:
Path Cost: 143
Number of nodes expanded: 503

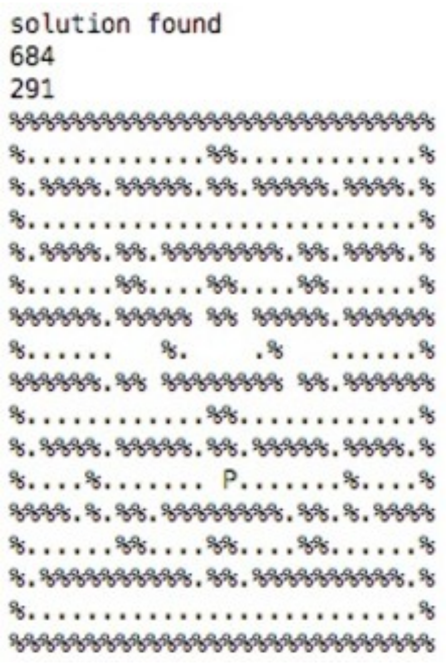
Input: mediumSearch

Solution:
Path Cost: 207
Number of nodes expanded: 3981



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% f  %        c%             %  7%     % %   %5 %
%    %%%% %%%%%% %  % % %%         %%%%  %  %%% %
%    %d%    % b% %  % %  %   %   %          %   %
%   e            %9     %%%% %   %% 6%%% %      %
% %%%%% %%%%%%  %%%% %           8%    %4 %%%%%%%
%g      %   j%       a% %%%  %%%%%%  % %  %   % %
%%% %%  % %%%%%%% %%%%% %    %  0 %% % %    %2% %
%    %  %      %  %      P% %%       %   %%%% % %
%h      %   %  %  %     %          %       1%   %
% % %%%    % %    %% %% %%% %% %  % %%%%%%%%%%% %
%   %i   %    %              %    %        3    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

5. Extra credit: Suboptimal search



```
solution found
684
291
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%............%%............%
%.%%%%.%%%%%.%%.%%%%%.%%%%.%
%..........................%
%.%%%%.%%.%%%%%%%%.%%.%%%%.%
%......%%....%%....%%......%
%%%%%%.%%%%% %% %%%%%.%%%%%%
%......   %.    .%   ......%
%%%%%%.%% %%%%%%%% %%.%%%%%%
%............%%............%
%.%%%%.%%%%%.%%.%%%%%.%%%%.%
%....%....... P.......%....%
%%%%.%.%%.%%%%%%%%.%%.%.%%%%
%......%%....%%....%%......%
%.%%%%%%%%%%.%%.%%%%%%%%%%.%
%..........................%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

For this question, our solution cost is 291, while the expanded nodes are 684, which is extremely small.

We still use the similar approach when dealing with “mediumSearch”. A graph is set up so we can try to solve a TSP problem instead of a maze solving problem. This could be a very fast process because there are only k*(k-1)/2 paths in total. But there are two biggest problems for “bigDots”, space complexity and the number of expanded nodes.

Before, by converting the problem into a graph, and to check repeated states, we need a k*2^k memory to store the states. Here, instead of storing which dots have been collected, we store how many dots have been collected. This results in the omission of a lot of unexplored states but can lead to a solution. Also, to reduce the computational cost by a huge amount, and because we only care about suboptimal solution, the heuristic is multiplied by a large constant, which made this algorithm kind of similar to greedy first search now because the heuristic takes much more weight that it should in A*.