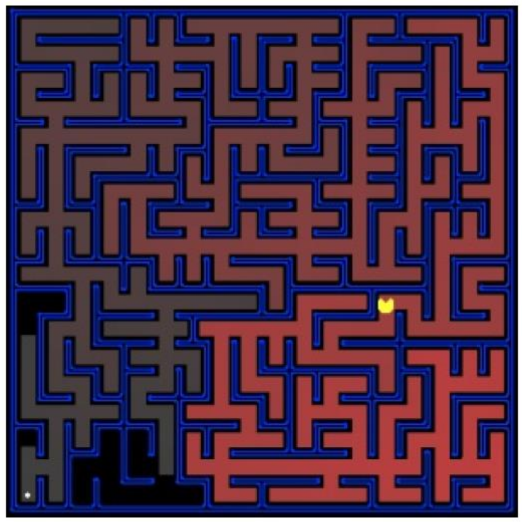
C M S C 1 7 0

MACHINE PROBLEM NO.1

"MAZE SEARCH HOORAY"

-



OVERVIEW

As stated in the beginning of the course, you are free to use any high-level programming language you are comfortable with. This includes (but is not limited to) Java, C/C++, and Python. The focus of this course is on problem solving, not programming, and the grading will primarily be based on the quality of your solutions and your analysis, as evidenced by your written report. To understand the requirements, be sure to carefully read all the instructions below.

PART ONE: BASIC PATHFINDING

To begin with, you will consider the problem of finding a path through a maze from a given start state to a given goal state. This scenario is illustrated in the figure above, where the start position is indicated by the "Pacman" icon and the goal state is a dot. The maze layout will be given to you in a simple text format, where '%' stands for walls, 'P' for the starting position, and '.' for the goal (see sample maze file). For this part of the assignment, *all step costs are equal to one*.

Implement *A* Search* in solving the different maze problems. Use the *Manhattan Distance* and the *Straight-Line* distance as heuristics. Compare the results afterwards.

Run A* search on the *small maze*, *medium maze*, *big maze*, and the *open maze*. For each problem instance, report the following:

a. The solution and its path cost (cost of one step (up, down, left, right) = 1);
b. Number of nodes expanded (number of nodes in the closed list);
c. Maximum size of the frontier (number of nodes in the open list).

You can display the solution by putting a '.' in every maze square visited on the path (see example solution to the big maze).

PART TWO: SEARCH WITH MULTIPLE GOALS

Now we consider a harder problem of finding the shortest path through a maze while hitting *multiple* goals (that is, you want to make the Pacman, initially at P, eat *all* the dots). Once again, in this part, we assume unit step costs.

Revise your code from Part 1 to deal with this scenario. This will require changing the goal test (have you eaten all the dots?) and the state representation (besides your current position in the maze, is there anything else you need to know?). Run A* from Part 1 on the *small search* and *tricky search*. For each maze instance, output your solution by modifying the maze file to number the goals in the order in which you reach them. Also, report the solution cost and number of nodes expanded (same with part 1).

HEURISTIC FORMULAS

For your guidance, here are the formulas for the heuristics that you will use in solving the problem:

Manhattan Distance	**Straight Line Distance**
abs(x1 – x2) + abs(y1 – y2) where: x1 = row index of starting location y1 = column index of starting location x2 = row index of goal location y2 = column index of goal location	dx = abs(x1 – x2) dy = abs(y1 – y2) D * max(dx, dy) where: x1 = row index of starting location y1 = column index of starting location x2 = row index of goal location y2 = column index of goal location D = cost per move (in this case, D is 1)

GENERAL GUIDELINES

1. This machine problem will be done in groups of three (3). Make sure that your groupmates are in the same section as you. Section C has 33 students, so there should be 11 groups in total. Section D has 32 students, so there must be 10 groups and one brave pair in total. Groups must only be a maximum of three students each. *No exceptions.*
2. Submit a written report explaining the performance of the heuristics in both Part 1 and Part 2, with screenshots of the solutions in PDF format.

3. In your report, include a page with a description of the contribution of each team member, with each team member signing on that page.
4. Email your output (codes + documentation) via a .zip folder to avgerman@up.edu.ph **ON OR BEFORE OCTOBER 23, 2017 MONDAY (11:59 PM)**.
5. Email subject: **[SUBMISSION] CMSC170_MP1**

DISCLAIMER

This machine problem was based on this assignment and was modified accordingly to Sir Tony Briza's specifications for his CMSC 170 class 1st semester, SY 2014 – 2015.