

Due: March 24 (Saturday) at 11:59 PM

**Objective:** To learn how to implement backtracking algorithms using a stack, and to gain experience using the two-dimensional array (Grid class) data structure.

**To start the project:** Download and unzip the file hw3.zip file at:

<http://www.cs.uni.edu/~fienup/cs1520s12/homework/hw5.zip>

**The Assignment:** (This is Project 6 of Chapter 17 on page 730. (See also Project 10 of Chapter 14 on page 602 of the text)

For this programming project, you are to write a program that solves the maze problem by using the recursive algorithm developed in Section 17.3.2 of the textbook. You should also review Sections 13.4 on Two-Dimensional Arrays (Grids) and Section 14.3.4 of the textbook where a non-recursive backtracking algorithm for the maze problem is outlined. The Grid class from chapter 13 can be found in the grid.py file contained in hw5.zip.

Your program should prompt the user for the name of a text file containing the description of a maze. Use a file format where

- the first line contains the number of rows in the maze,
- the second line contains the number of columns in the maze, and
- the remaining lines contain the maze where '\*'s are walls, ' 's (spaces) are paths, and 'P' is the starting Parking lot, and 'T' is the target mountain Top. The file maze1.txt contains the following example.

```
10
20
*****
***   *****   *****
***   *****   *****
P           *****
***   **   *****   *****
***   **   *****   *****
***   **   *****
*       *           T
*****
*****
```

Your program needs to:

- determine whether a solution is possible (SUCCESS) or not (FAILURE).
- print the maze as above, with '.'s (periods) along the unsuccessful side-trips and 'X's along the successful path from the 'P' to the 'T'.

**For extra credit,** your program may also do one or more of the following:

- print a list of moves needed to get from the 'P' to the 'T' with unsuccessful side-trips eliminated if a solution to the maze is possible.
- print the maze as above, with '.'s (periods) along the unsuccessful side-trips and '+'s along the **shortest** successful path from the 'P' to the 'T'.

**Implement AND fully test your maze solving program.** *Part of your grade will be determined by how well you test your program using different mazes.* You will be required to submit all of your test mazes with your program for grading. Use names of the form maze1.txt, maze2.txt, maze3.txt, etc. for the test mazes.

**Submit all data files (maze#.txt) and program files as a single zipped file (called hw5.zip) electronically at**

[https://www.cs.uni.edu/~schafer/submit/which\\_course.cgi](https://www.cs.uni.edu/~schafer/submit/which_course.cgi)