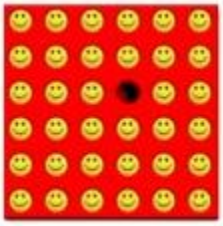
CS480/580 Introduction to Artificial Intelligence
Assignment 1

Total Points: 100
Due Date: 9/29/2020

Pegboard Game



In this assignment, you are asked to implement a working solver for the pegboard game. Pegboard problems are single-player games played on a grid, in which moves are made by successively jumping and removing pegs from the pegboard. A peg can jump an adjacent peg if there is a slot adjacent to that peg in the opposite direction – horizontally or vertically. Diagonal jumps are not allowed. After a peg has been jumped, it is removed from the board (and possibly eaten). A typical objective of this problem is to begin with a full pegboard from which one peg has been removed and determine a sequence of jumps which will result in one peg remaining. The pegboard game has been a challenging problem for a human being. The following is the solution of a 6x6 pegboard game.

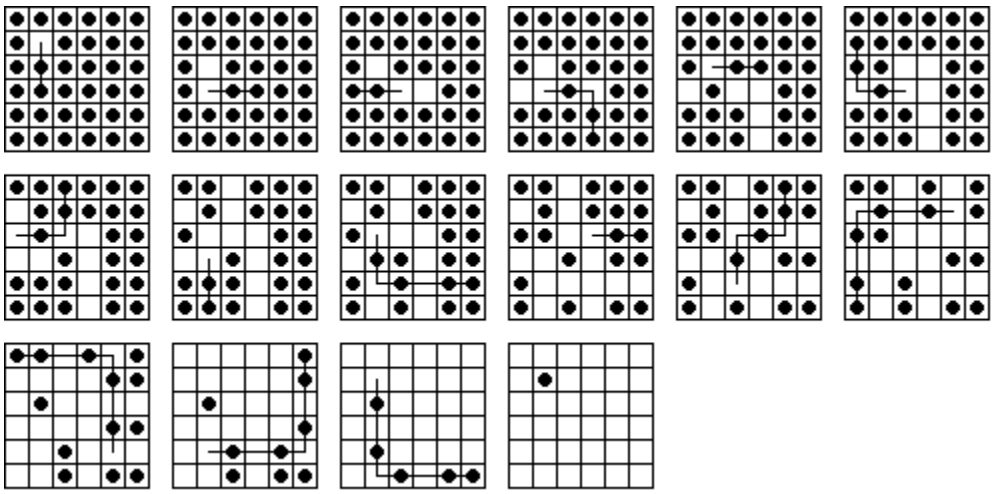


Image from http://recmath.org/pegsolitaire/

Setup of the puzzle

The pegboard game is played on an m x n grid board. For example, the initial state of a 4x4 pegboard can be represented as

```
0111
1111
1111
1111
```

A possible goal state can be

```
0000
0100
0000
0000
```

Heuristics

It is very helpful if you can play on the pegboard by yourself and figure out some strategies that lead to the solution. There are a variety of methods to estimate how "promising" a given pegboard state is. Particularly, one can notice that leaving pegs isolated in the edges is a bad strategy and thus one should always try to concentrate the pegs around the center. The heuristics functions are used to measure how well the pegs are concentrated around the center. You can consider using one of the following heuristics or implement your own.

- Manhattan. This heuristic's value of a board is the sum, for each peg on the board, of the Manhattan distance from the center.
- Exponential Distance. If H,V represent the horizontal and vertical distances from the center respectively, the heuristic's value is $2^{max(H,V)}$.
- Connected Components. The connected components favor states in which no two concentrated groups of pegs are so far away from each other that they will never meet.

Programming guide

Your program should be able to represent the pegboard as the state in the above format. In addition to the functions of handling the

goal(state): returns True if state equals the state with exactly 1 peg

successor(state): given a state, returns all possible states from the current state

heuristic(state): given a state, returns the heuristic function value

Programming Tasks

1. You need to write programs to implement breadth-first search and depth-first search to generate solutions for given 4x4, 5x5, 6x6, 7x7, 8x8, 9x9, and 10x10 pegboard puzzles.

2. Implement a heuristic function. Explain your heuristic in your comments in under 500 words.

3. Implement programs using greedy-best search and A* search algorithms based on your heuristic to generate the solution for given 4x4, 5x5, 6x6, 7x7, 8x8, 9x9, and 10x10 pegboard puzzles.

Analysis

For each of the search algorithm (BFS, DFS, greedy-best, A*), compare the solutions found, the number of search nodes explored, the computational time for given 4x4, 5x5, 6x6, 7x7, 8x8, 9x9, and 10x10 pegboard puzzles. Justify your analysis using the completeness, optimality, time complexity, space complexity analysis we discussed in class.

What to Hand in

1. Well documented codes implementing breadth first search, depth first search, greedy search, and A* search. A README file should provide instructions on how to compile and execute the code.
2. Provide the solutions generated by your programs using BFS, DFS, Greedy search and A* search on different pegboard puzzles. If no solutions found, state no solutions.
3. Analyze results of the BFS, DFS, Greedy Search, and A* search algorithms. Analyze your results, for example, try to find out when your algorithms fail to come up with a solution.

Please turn in the program and the analysis before the assignment due date.

Hints:

1. Good data structure representation can make your life easy.
2. You may want to start from a very simple pegboard such as a 3x3 problem.

Bonus (10 pts)

You may want to challenge your program with one of these historic pegboard games.