CS160 Lab #9: Data Structures, Search, and Programming

General

Read through the "Background" material below and the online tutorials, then download the Lab #9 question set and answer the questions. Turn in the questions using the instructions posted on the class web site.

For ALL Word processing documents, you must submit your documents in one of the following formats: MS-Word (NOT Works), RTF (most word processors can save in this format), or Open Document (used by the freely available Open Office suite). They will be returned ungraded if submitted in any other format.

Concepts

This lab will introduce you to the concepts of ***data structures*** and ***search***. The focus is on abstract data types and continues your study of algorithms. We will also use the "Alice" programming environment tutorials for part of this lab for more experience with programming.

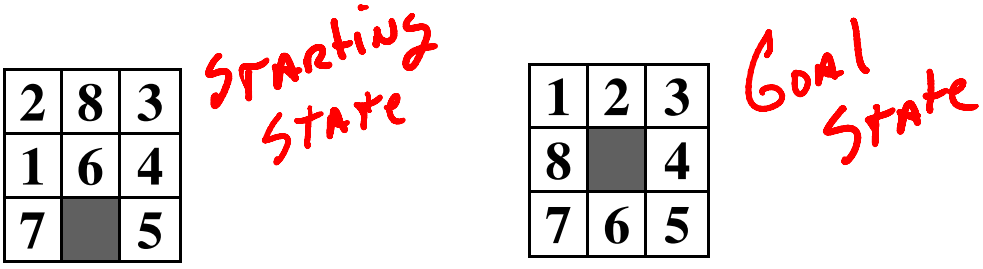
Background

Note: Original source of the background contained below is from the "CS160 Worksheets" by Daniel Balls of the CS department at Oregon State University; updated and revised by Mitch Fry (CS, Chemeketa Community College).

Search Methods

Some situations present problems that have many solutions, but only one that is optimal. In such a situation a searching strategy may be helpful in finding the optimal solution. To illustrate problem solving by searching, we'll consider the 8-puzzle.

The 8-puzzle is tile game consisting of nine squares—eight numbered tiles and one 'empty' square (see square below, left which was adapted from Zhu, S. C., http://www.stat.ucla.edu/~sczhu/Courses/UCLA/Stat_232B/Handouts/Introduction.pdf.). The goal of the puzzle is to use the free space to move the tiles so that the numbers are in order starting at the top left and proceeding around the square until the 8, which is right below the 1 (see square below, right). When the numbers are in this order, we have found a solution.



The puzzle on the left represents an 8-puzzle in a non-solution state, while the puzzle on the right represents the solution, or objective, of an 8-puzzle.

An optimal solution consists of one that consists of the fewest slides. To track the progress toward the optimal solution, we'll implement a penalty rule system. Each time we slide a tile we'll be assessed one point. Further, for each number that is out of its solution position, we will assess one penalty point. So if the 8 – puzzle we want to solve is the one on the left above,

the initial position has a score of 4 points—we've made no slides, and there are four numbers (2, 8, 6, and 1) out of place.

First, we consider the three possible first moves and the number of points for each of the resulting states, as shown below. We'll continue to search from the puzzle with th e lowest point total, in this case the puzzle with 4 points.

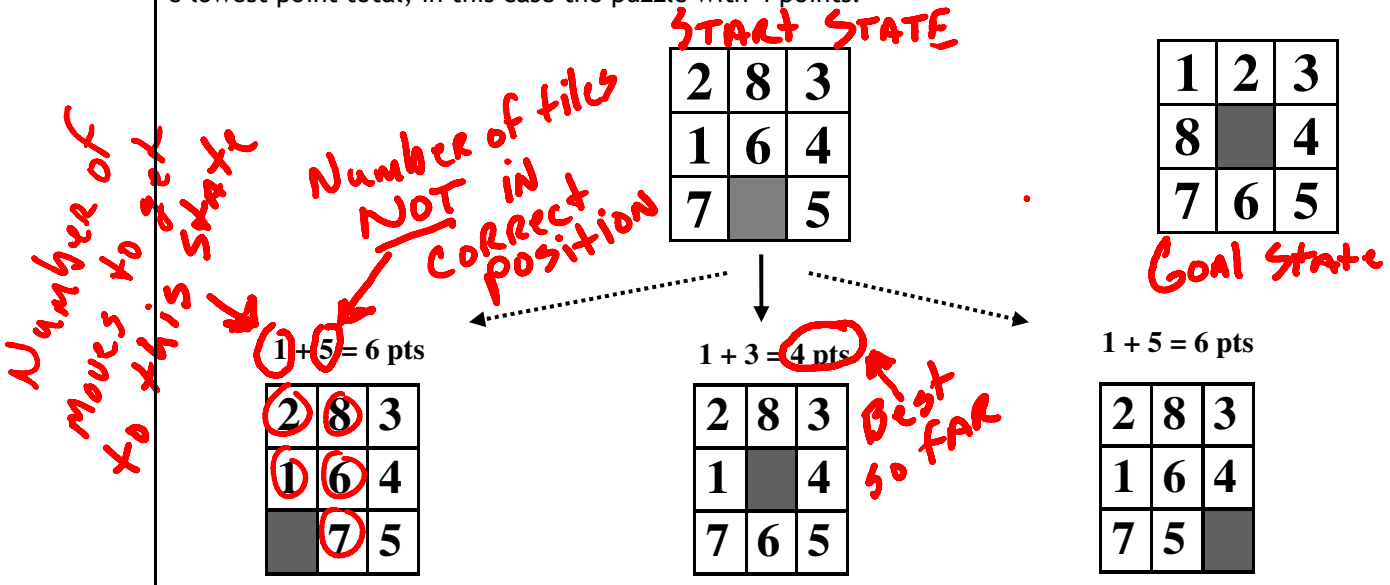
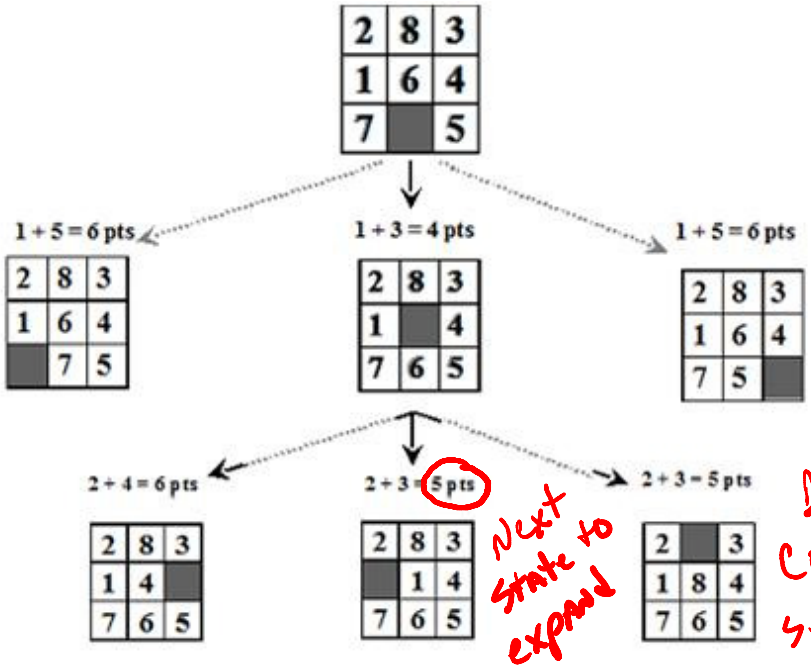
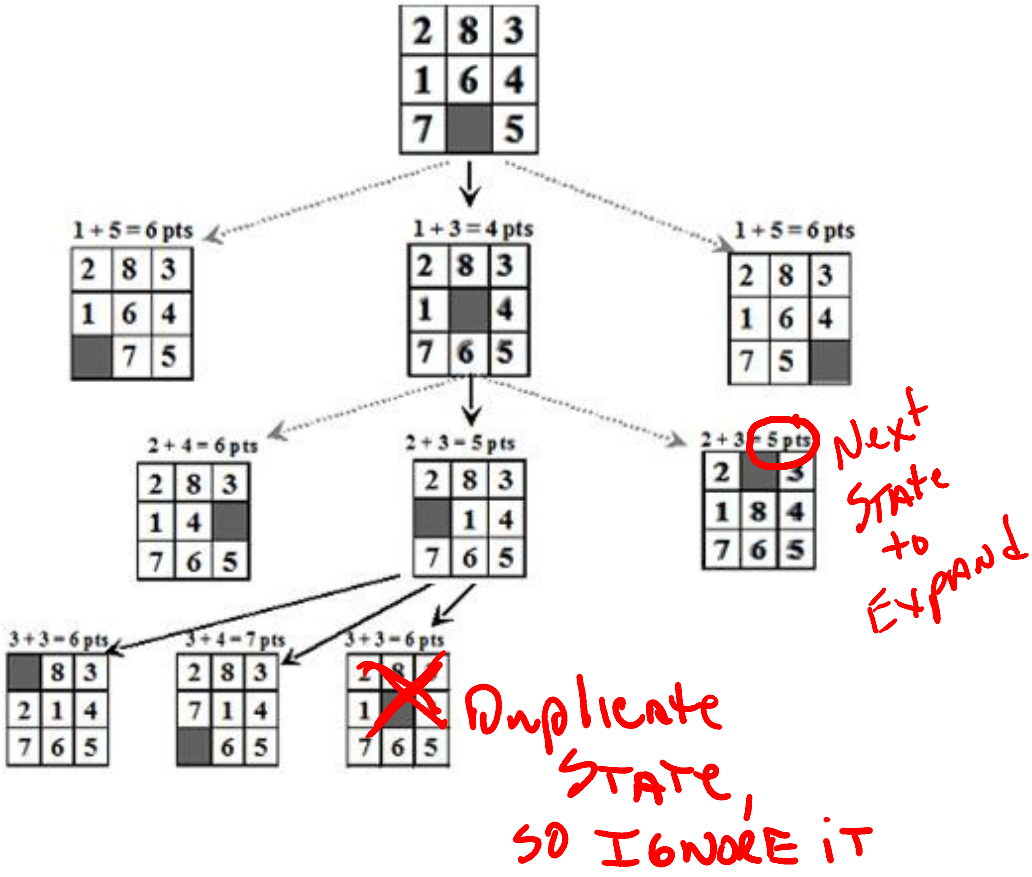


Diagram note: *a shaded puzzle represents one whose 'next steps' have been expanded. Solid arrows represent the current 'puzzle path' we are currently exploring. Dotted arrows represent paths to puzzles we have not currently chosen, but may choose in the future if they become more desirable than the current path.*

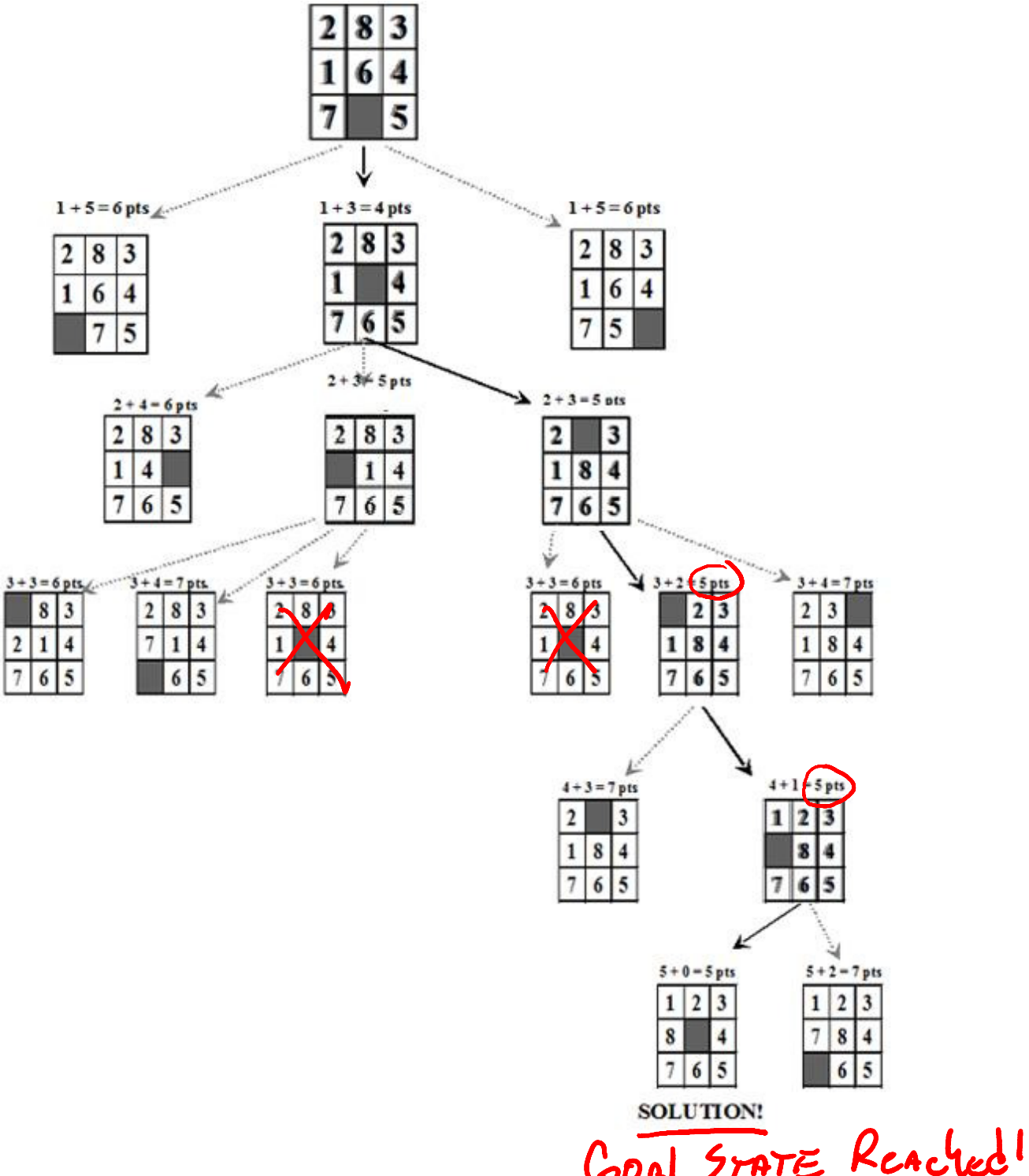


Since two puzzles tie for the lowest number of points (i.e. two of them have only five points), we'll choose one of the two arbitrarily, the one with the empty square in the middle row.



Searching this path leads to three puzzles with point totals of six (two have this many points) and seven. At this point if we were using the greedy method, we would follow the path of one of the puzzles having six points. However, according to the search method we will take a step back and survey all possible remaining paths and then choose from them the one that is most desirable. Currently there are seven puzzles that have not yet been expanded. Each of these puzzles has five, six or seven points. There is only one puzzle with five points and we'll expand that puzzle next, as it is currently the most desirable path.

As shown on the next page, by following this path and implementing the search strategy, the optimal solution to the puzzle can be obtained in five slides, indicated in the diagram by the solid black arrows.



The key principle of this searching method is always following the path that appears to be leading to the lowest-cost solution. If the algorithm is following a promising search path, but at some point the path seems to be inferior to another path through the search tree, the algorithm should switch focus and pursue the path that now seems to be better.