

Problem Set 2

Handed Out: 2/17/2009

Due: 3/2/2009

1. [Search Spaces]

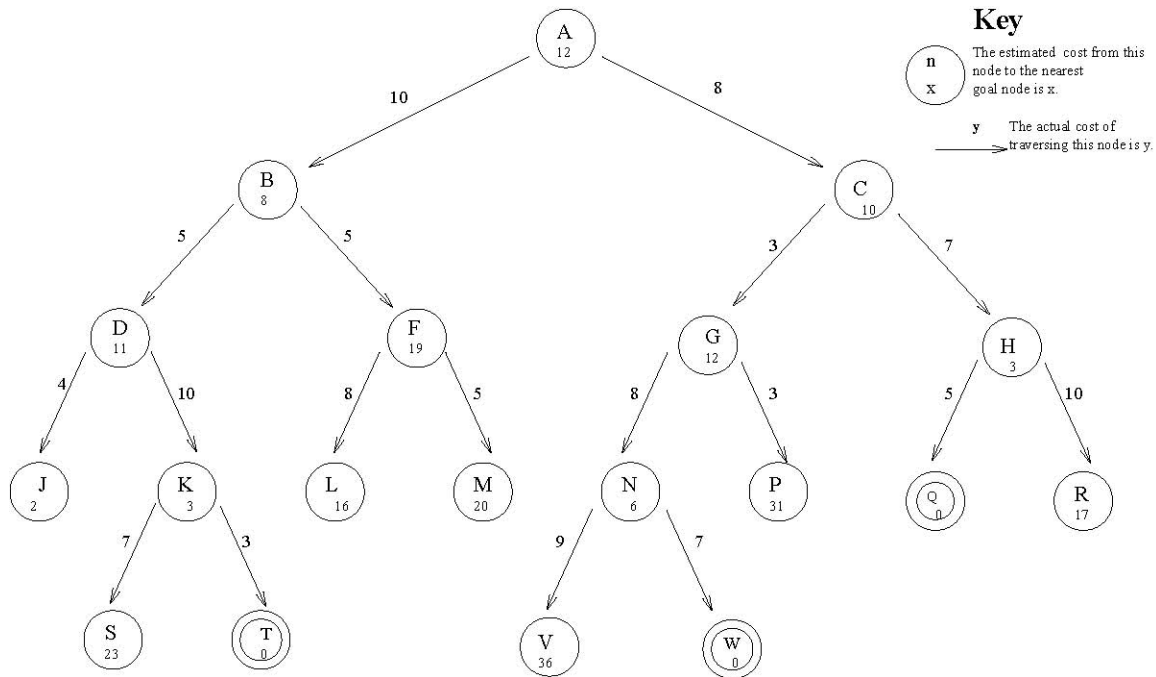


Figure 1: Search Tree

For parts (a)-(e), consider the search tree in Figure 1. The initial state is at the top, and goal states are represented by double circles. Note that the edges are directed. The search method will determine which goal is reached (if any). For each of the search strategies listed below, form an ordered list of the states explored¹ and indicate which goal state is reached (if any). Use an ascending alphanumeric ordering (A < B < ... < Z) of state labels to break any ties which may occur. Please show your work at each step of the search algorithm.

- (a) Depth-First Search
- (b) Best-First Search
- (c) Heuristic Search
- (d) For the graph in Figure 2, use a Dynamic Programming algorithm to find the shortest path from node A to node E. Fill in the Φ matrix as shown in class, and draw arrows to show the path.

¹ A state is considered explored when it is removed from the queue.

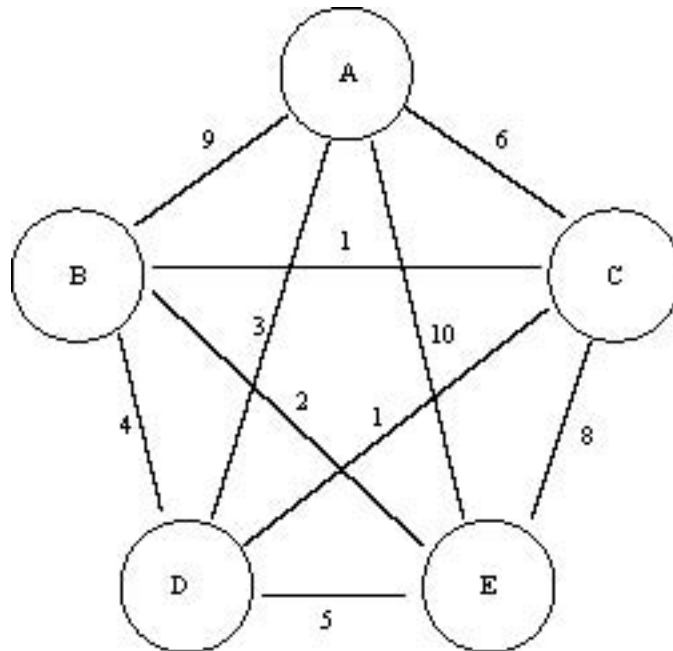


Figure 2: Graph with five nodes.

2. [Search Space Formulation]

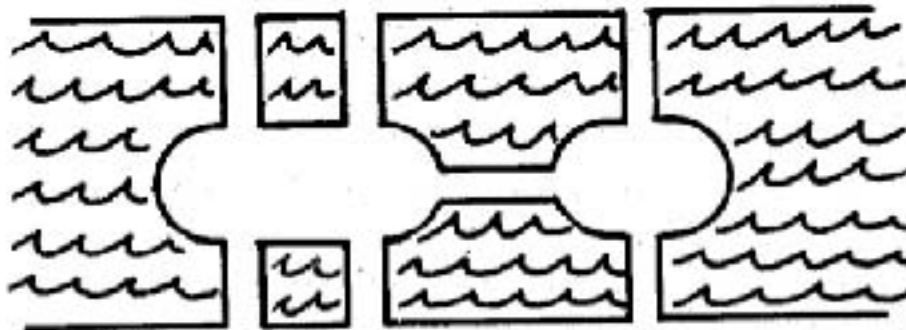


Figure 3: The Seven Bridges of Königsberg

The Seven Bridges of Königsberg problem is as follows: In figure 3, traverse each of the seven bridges once and only once and return to the point from which you started.

Formulate this task as a state space search problem:

- (a) What does each state in the search represent? What are the initial and final states?
- (b) What does an edge between states represent? What does it mean for two states in this space to be connected by an edge?
- (c) Are there reasonable heuristics to help search through this space? If so, what are they?
- (d) For this particular instance of the problem, what will the result of the search be?

3. [Triangle Peg Solitaire]

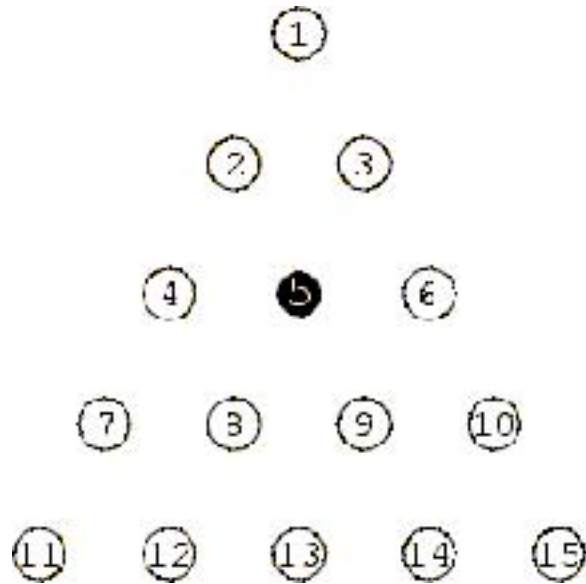


Figure 4: Triangle Peg Solitaire

This problem is a machine problem. You will be asked to solve the Triangle Peg Solitaire problem. The game is played on the board shown in Figure 4, consisting of a number of holes of which all but one (in this case hole #5) are initially filled with pegs. The goal is to remove all pegs but one by jumping pegs from one side of an occupied peg hole to an empty space, removing the peg which was jumped over.

(a) Formulate this task as a state space search problem as in Problem #2. In this case, you will not be required to find reasonable heuristics or solve the problem by hand.

(b) Implement an algorithm that will search through a generic state space with a particular search method as described in class.

(c) Apply this algorithm to the Triangle Peg Solitaire problem as formulated in part (a). Your program should output a series of legal moves that when applied to the board in Figure 4 will leave only one peg remaining on the board. These moves should be output in the following format:

```
Move peg at hole i to hole j Move  
peg at hole k to hole m  
...
```

What is the final state of the board after these moves? Is it possible to find a different final state by using a different search method?

(d) Suppose that the board in Figure 4 is modified to have hole #4 vacant instead of hole #5. Repeat part (c) for this modified board.