CS121 – Introduction to Artificial Intelligence – Winter 2011

Homework #2
(Blind and Heuristic Search)
Out: 1/17/11 — Due: 1/25/11 (at noon)

How to complete this HW: First copy this file; then type your answers in the file immediately below each question; start each question on a separate page, write your name on every page. Finally, print this file and return it in the drawer marked CS121 of a file cabinet located in the entryway of the Gates building next to office Gates 182 (see red arrow in http://ai.stanford.edu/~latombe/cs121/2011/map.jpg) no later than Tuesday 1/25 at noon.

Your name: ……………………………………………………………………………………

Your email address: ………………………………………………………………………

Note on Honor Code: You must NOT look at previously published solutions of any of these problems in preparing your answers. You may discuss these problems with other students in the class (in fact, you are encouraged to do so) and/or look into other documents (books, web sites), with the exception of published solutions, without taking any written or electronic notes. If you have discussed any of the problems with other students, indicate their name(s) here:
………………………………………………………………………………………………………
Any intentional transgression of these rules will be considered an honor code violation.

General information: Justify your answers, but keep explanations short and to the point. Excessive verbosity will be penalized. If you have any doubt on how to interpret a question, tell us in advance, so that we can help you understand the question, or tell us how you understand it in your returned solution.

Grading:

Problem#	**Max. grade**	**Your grade**
I	25	
II	25	
III	25	
IV	25	
Total	100	

Name ………………………………………………………………..

I. Toy railway (25 points)

A toy-railway set contains four types of track pieces shown in Figure 1. There are 12 straight pieces, 16 curved pieces, 2 fork-1 pieces and 2 fork-2 pieces. Each straight and curved piece has a *hole* at one end and a *peg* at the other end. A fork-1 piece has one hole at one end and one peg at each of the other two ends. A fork-2 piece has one peg at one end and one hole at each of the other two ends. Two pieces can be *connected* together by inserting a peg of one into a hole of another. The pieces fit together exactly with no slack. The curves and fork pieces can be flipped over; hence, they can curve in either direction. Each curve subtends a 45-degree arc.

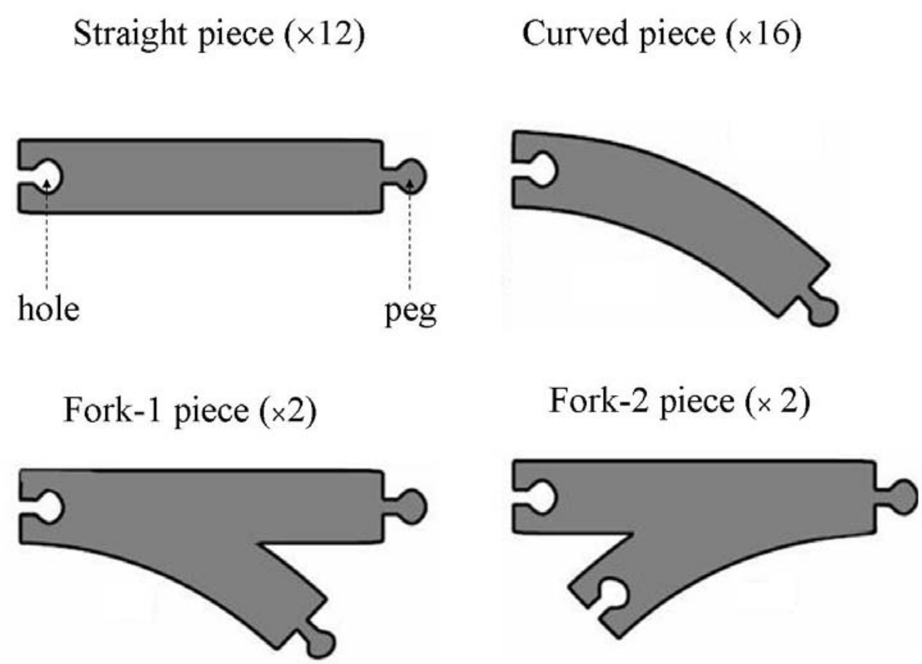


Figure 1. The pieces of the toy-railway set

The task is to connect **all** those pieces into a **single** railway track that has no **open ends** where a train could run off and no **overlapping** tracks.

1. Give a precise formulation in words of the task as a search problem, that is, describe the state space, the successor function, the initial state, and the goal test.
   [Hints and comments:
   - If this helps, you may call a peg that has not been inserted yet into a hole, an *open* peg.
   - Choose your successor function to limit the branching factor; if your branching factor is above 10, you will not get full credit. State explicitly how many successors are contributed by each piece of a given type.
   - You do not need to provide all details of the goal test; just state what need to be tested.
   - Ignore step cost.]
2. Among the following blind search methods – breadth-first, depth-first, depth-limited, iterative deepening – which one would you choose? Why?

Name ………………………………………………………………..

3. Explain briefly why removing any one of the fork pieces from the railway set of Figure 1 would make the problem unsolvable. Derive a general *necessary*, but not *sufficient*, condition for any given railway set (using the 4 types of pieces shown in Figure 1) to have a solution.

4. Give an upper bound on the number of nodes generated by either breadth-first or depth-first search for your formulation at Question 1. [This question will earn you points only if your formulation at Question 1 is correct. Your bound need not be very tight, but your grade will still depend on how tight it is.]

Name ……………………………………………………………………..

II. Vehicles on a grid (25 points)

A group of n vehicles, identified by 1, 2, ..., n, can move in an $n \times n$ grid of squares. Initially, each vehicle i is located in square $(i,1)$ [meaning on i^{th} column and 1^{st} row]. Its goal is to move to square $(n-i+1,n)$ [hence, the vehicles must move to a reverse order in the n^{th} row of the grid].

At each step, each of the n vehicles can move by one square up, down, left or right, or stay in the same square. If a vehicle does not move, one adjacent vehicle (but no more than one) can hop over it. No two vehicles are allowed to be in the same square at the end of any given step.

1. What is the approximate size of the state space for this problem?

   (i) n^2 (ii) n^3 (iii) n^{2n} (iv) n^{n^2}

2. What is the approximate branching factor of the search tree?

   (i) 5 (ii) $5n$ (iii) 5^n

3. Suppose that vehicle i is located at square (x_i,y_i). What is the minimum number h_i of steps that this vehicle must perform to reach its goal square assuming that there are no other vehicles on the grid? [Express h_i in mathematical form as a function of x_i, y_i, i, and n.]

4. Is $\min\{h_1,...,h_n\}$ an admissible heuristic for the problem of moving all n vehicles to the goal squares (when the cost of a solution is the number of steps to reach the goal)? Is $h_1+...+h_n$ an admissible heuristic? Justify your answers.

Name ……………………………………………………………………..

III. Admissible Heuristic Function (25 points)

Consider the 8-puzzle problem.

1. Assume that the goal state is the following state:

1	2	3
4	5	6
7	8	

   In any state *s*, for any non-empty tile i = 1 to 8, the number n_i of permutation inversions is the number of tiles $j < i$ that appear after tile i (as defined in class). The total number of permutation inversions of *s* is $P = \Sigma_{i=1,\ldots,8}\, n_i$.

   Let $h(N)$ be the heuristic function that evaluates to the total number of permutation inversions in the state of node N. Is h admissible? If yes, prove it. If not, show a counter-example.

2. Assume now the goal test only specifies the number of each of the 3 tiles in the top row (hence, there are multiple goal states). For instance, the goal may be:

1	2	3
x	x	x
x	x	x

   where each "x" stands for any tile other than 1, 2, or 3 (all tiles marked "x" are distinct).

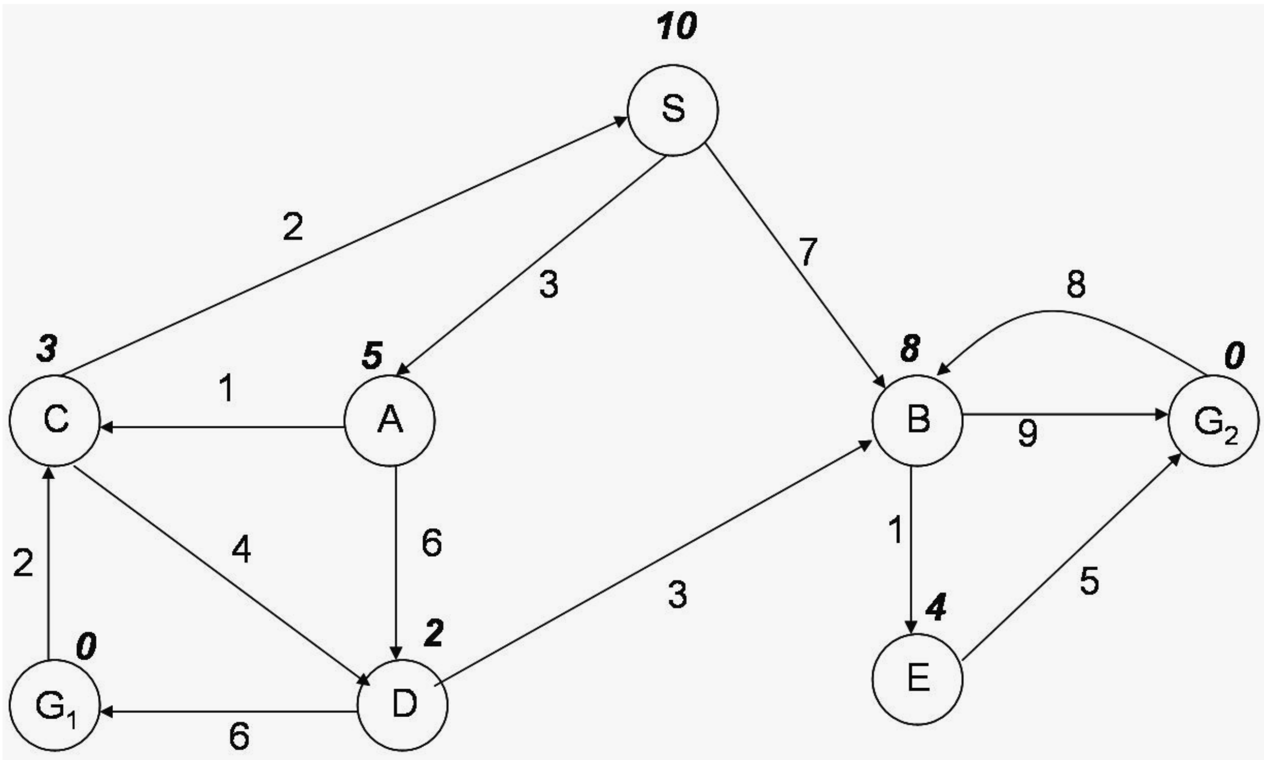
   Define two admissible heuristic functions for such a goal condition (other than trivial admissible functions such as h=0 for all states, or h=1 for all non-goal states and h=0 for all goal states).

3. Assume that the goal test requires that the sum of the tiles on the first row must be exactly 9. Define an admissible heuristic function for such a goal condition (other than a trivial function). [Hints: Count how many combinations of tiles in the first row satisfy the goal test. Try to use the results of Question 2.]

Name …………………………………………………………………..

IV. Heuristic search (25 points)

Consider the following state graph:



The state space consists of states S, A, B, C, D, E, G_1, and G_2. S is the initial state and G_1 and G_2 are the goal states. The possible actions between states are indicated by arrows. So, the successor function for state S returns {A, B}; for A it returns {C, D}, etc... The number labeling each arc (roughly at the mid-point of the arc) is the actual cost of the action. For example, the cost of going from S to A is 3. The number in bold italic near each state is the value of the heuristic function *h* at that state. For example, the value of *h* at state C is 3.

1. Fill the following table (on next page) with the nodes successively added to the fringe by the best-first search algorithm using the evaluation function $f(N) = g(N) + h(N)$, where $g(N)$ is the cost of the path found from the initial node to node *N*. The algorithm does not check if a state is a re-visited one or not (hence, there may be several nodes with the same state in the search tree). It terminates only when it removes a goal node from the fringe. The states produced by the successor function are always ordered in alphabetic order. In the rightmost column (#exp), indicate the order in which nodes are expanded (i.e., are removed from the fringe). If a node is not expanded, leave the corresponding cell empty. The first line of the table is filled for you. It is possible that you do not need all the rows in the table. In this problem, you don't have to justify your answers.

Name …………………………………………………………………..

N	State	$g(N)$	$h(N)$	$f(N)$	#exp
1	S	0	10	10	1
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

2. Is the heuristic function *h* defined by values provided in the figure admissible? How do you know? How does this affect the search?