CS 121, Summer 2009
Homework #2

Out: July 1, 2009
Due: July 13, 2009

How to complete this Homework: Your answers can be typed or carefully hand-written. Please begin each problem on a new page and make sure your name is written on each page of your assignment. Print out this cover sheet and fill in your name and email address, as well as any people you collaborated with. Staple all of your work together and turn in at the beginning of class, July 13, 2009. If you will not be in class you can turn it in prior to class under the door of Gates 132 (with the time of submission written on the homework). Late homeworks will not be accepted.

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, so long as your final submission is prepared on your own without referring to any notes taken during such collaboration. 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.

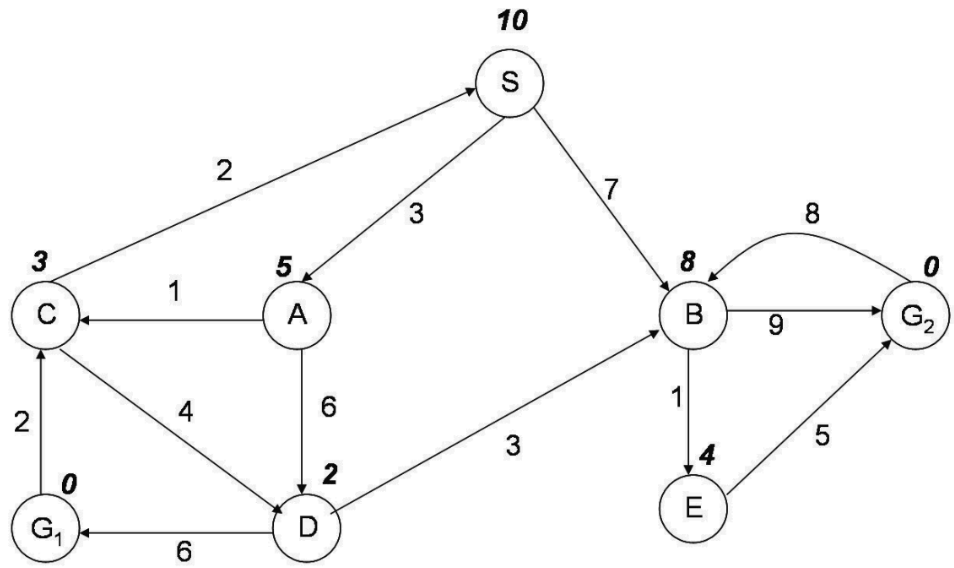
General Information: Justify your answers, but keep explanations short and to the point. Excessive verbosity will be penalized. If you have any doubt on how to interpret a question, tell us in advance, so that we can help you understand the question, or tell us how you understand it in your returned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

Grading:

Problem #	Max. Grade	Your grade
1	20	
2	20	
3	20	
4	20	
5	20	
Total	100	

---

1. **Heuristic Search [20 points]** Consider the following state graph:



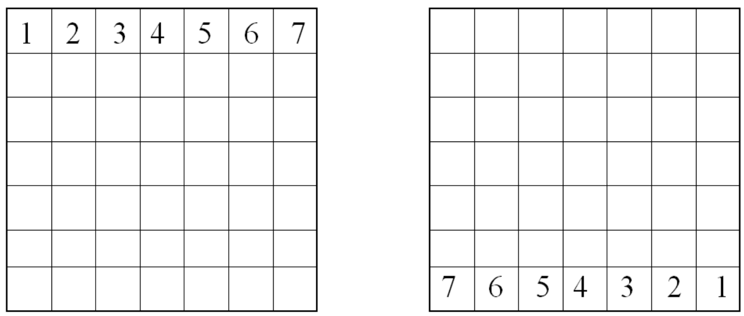
The state space consists of states S, A, B, C, D, E, G_1, and G_2. S is the initial state and G_1 and G_2 are the goal states. The possible actions between states are indicated by arrows. So, the successor function for state S returns $\{A, B\}$; for A it returns $\{C, D\}$, etc .... The number labeling 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.

(a) **[15 points]** Copy the following table (with as many rows as necessary) in your submission and fill it in 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. In this problem, you don't have to justify your answers.

N	State	$g(N)$	$h(N)$	$f(N)$	# exp
1	S	0	10	10	1
2					
3					
$\vdots$					

(b) **[5 points]** Is the heuristic function h defined by values provided in the figure admissible? How do you know? How does this affect the search?

2. **Vehicles on a grid [20 points]** A group of n vehicles, identified by $1, 2, \ldots, n$, can move in an $n \times n$ grid of squares. Initially, each vehicle i is located in square $(1, i)$ [meaning on 1st row and i-th column]. Its goal is to move to square $(n, n-i+1)$ [hence, the vehicles must move to a reverse order in the n-th row of the grid].



Initial state Goal state

Example grid with $n = 7$

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.

(a) **[3 points]** What is the approximate size of the state space for this problem? Briefly explain your choice.

   i. n^2
   ii. n^3
   iii. n^{2n}
   iv. n^{n^2}

(b) **[3 points]** What is the approximate branching factor of the search tree? Briefly explain your choice.

   i. 5
   ii. $5n$
   iii. 5^n

(c) **[6 points]** Suppose that vehicle i is located at square (r_i, c_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 r_i, c_i, i, and n.]

(d) **[8 points]** Is $\min\{h_1, \ldots, h_n\}$ 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 $h_1 + \ldots + h_n$ an admissible heuristic? Justify your answers.

3. **Search with comparison functions [20 points]** Sometimes there is no good evaluation function for a problem, but there is a good comparison method: a way to tell whether one node is better than another, without assigning numerical values to either. Assume that we have a search problem with such a comparison method.

   (a) **[10 points]** Can we perform best-first search in this search space? Explain your answer, detailing how we would do this if you think it is possible, or why we can't if you think it is impossible.

   (b) **[10 points]** Can we perform A* search in this search space? Explain your answer, detailing how we would do this if you think it is possible, or why we can't if you think it is impossible.

4. **Admissible heuristic function [20 points]** Consider the 8-puzzle problem.

   (a) **[4 points]** 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. The total number of permutation inversions of s is $P = \sum_{i=1,\ldots,8} n_i$.

   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.

   (b) **[8 points]** 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 the trivial function $h = 0$).

(c) **[8 points]** 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 the trivial function $h = 0$). [Hints: Count how many combinations of tiles in the first row satisfy the goal test. Try to use the results of part 4b.]

5. **Approximately optimal search [20 points]**

The two objectives of finding a solution as quickly as possible and finding an optimal solution are often conflicting. In some problems, one may design two heuristic functions h_A and h_N, such that h_A is admissible and h_N is not admissible, with h_N resulting in much faster search most of the time. Then, one may try to take advantage of both functions.

(a) **[15 points]** A best-first search algorithm called A^*_ϵ uses the evaluation function $f(N) = g(N) + h_A(N)$. At each iteration, A^*_ϵ expands a node N such that $f(N) \leq (1+\epsilon) \times \min_{N \in \text{FRINGE}}\{f(N)\}$, where ϵ is any strictly positive number. What can you say about the cost of the solution returned by A^*_ϵ?

(b) **[5 points]** Explain briefly how A^*_ϵ can use the second heuristic function h_N to reduce the time of the search. What tradeoff is being made in choosing ϵ?