CSE 473
Artificial Intelligence

Problem Set 1
Due 9:30am on 4/16

Please work independently on these problems. But if you do discuss them with another person, please note the name of the person and nature of the exchange in your answer. Similarly, if you get answers, hints or code from the Web, note the resource and where you found it in your answer.

You will be graded on the clarity (the ideal is a short, to-the-point answer) and neatness of your answers in addition to the technical correctness. Programming assignments will be judged in part in terms of the clarity of your solution; this includes commenting, formatting, description of approach, and demonstration that the code functions correctly). If the grader has to struggle to convince him- or herself that the code works correctly, this will be reflected in a less-than perfect score. Finally, if any problem appears underspecified, make reasonable assumptions that will let you solve it while preserving the spirit of the exercise; you will be graded on the choices you make.

1. From the textbook, R&N problem 2.9.
2. R&N problem 2.12a
3. R&N problem 3.9a (Note: normally I recommend typing homework to be turned in, but you'll likely waste too much time if you try to draw the state space with a computer drawing program. Paper and pencil is fine for this one.
4. R&N problem 3.9c
5. State the Symbolic Integration problem formally in terms of:
    - set of states
    - operators
    - start state
    - goal state/test

   It may help to think how people solve such problems and what your Calculus 101 teacher would consider a good answer.

6. The traveling salesman problem (TSP) can be posed with a set of cities C= {c1 … cn} and a distance function d(ci, cj), which returns a positive integer for each pair of distinct cities. The objective is to find the shortest path that visits each city exactly once.
    a. Specify the TSP as a search problem
    b. A powerful, admissible heuristic for TSP is based on estimating the remaining cost for completing a partial tour with the sum of the link costs for the minimum spanning tree connecting the graph of cities not yet in the tour. (The MST can be computed relatively quickly, i.e. in O(E lg E) time, using Kruskal's algorithm). Show how this heuristic can be derived from a relaxed version of the TSP. The key to this assignment (b) is making sure

that in (a) you specify the search problem in such a way that here in (b) you may simply say that you are relaxing (eliminating) parts i, iv, and viii of part (a) to get the new problem (minimum spanning tree). You may even find it easier to start the whole problem by writing MST as a search problem and then add constraints to it to yield part (a). Clear and concise answers appreciated.

7. Implement A* search (hint: you may start from scratch or use the java code that we provide on the Assignments page on the class web site, http://www.cs.washington.edu/education/courses/cse473/04sp/hw/, but please do not search the Web for A* search or 8-puzzle code) and apply it to the 8-puzzle. Turn in your commented code and the code for each heuristic.
8. Implement the following heuristics:
    a. $h_a(n) = 0$; i.e. leading to breadth-first search
    b. $h_b(n)$ = the number of misplaced tiles (excluding the blank tile).
    c. $h_c(n)$ = the sum of the distances of the tiles from their goal positions (excluding the blank tile).
    d. OPTIONAL: implement pattern databases, devise your own heuristic, compute the max of all the heuristics above, or try a non–admissible heuristic.

    Generate a set of 100 random initial boards and test A* running each heuristic on each of the 100 problems. Turn in a table of the average number of nodes explored for each heuristic. (OPTIONAL: turn in a more detailed table (with depth as an axis) like that of R&N figure 4.8).