CSCI 561 – Spring 2005 – Homework #1 – Due 2/08/2005 in class

Problem 1 [40%]: Artificial agents

Let's do a little bit of research on existing artificial agents and robots. For each of the following, please give us the PEAS (see textbook p. 38, 2nd ed) description which *you believe* the engineers who created the machine came up with. For each one, you will need to search on the web for what it can and cannot do, what it has in terms of sensors and actuators, what assumptions it is making about the world. In your answer, clearly mark out 4 bullets for each of P, E, A, and S. Each is worth 2% of the total grade.

a) Mars exploration rovers by NASA
b) Honda humanoid robot ASIMO
c) Robot soccer
d) Computer chess player
e) Automated Web Crawler

Problem 2 [60%]: Programming assignment: search

Please write a program that will use a search algorithm to find the best way for a robot to clean a messy house. The houses will be mapped as shown below.

Y \ X	0	1	2	3	4	5
0	1	1	1	1	1	1
1	1	0	0	1	2	1
2	1	1	0	1	0	1
3	1	0	0	0	0	1
4	1	2	0	0	2	1
5	1	1	1	1	1	1

RULES:

The houses will always be rectangular. The boxes marked with 1's are filled with walls, tables, or couches and are impassable, those marked with 0's are empty and are passable, and those marked with 2's are messy and need to be cleaned. The border of the houses will always be filled with 1's. When the robot moves over a messy box (marked with a 2) it becomes clean. The robot cannot move diagonally, it can only go up, down, left, or right into boxes marked with a 0. The robot starts in the starting box (whose coordinates will be given to you as input) and must clean all the messy boxes and then go to the charging station (whose coordinates will also be given). The robot's starting box and the charging station's box will always be in boxes marked with a 0. It is not guaranteed that a solution always exists.

Questions:
a) [4%] What is the PEAS description of this problem.

b) [3%] Which of the uninformed search algorithms studied in class do you think will work best [justify why with a little analysis/table]?

c) [3%] Which of the informed search algorithms studied in class do you think will work best [justify why with a little analysis/table]?

d) [40%] Let's see that code! Program an algorithm to solve the proposed problem, using the following conventions and either breadth-first or depth-first search:

CAUTION: please follow these simple rules

- write your name and email on the top of all files.
- do not expect any command-line arguments.
- make sure your code compiles on aludra.usc.edu using g++
- provide a Makefile or indicate in a README how to compile your code.
- your program should read one text file named "input.txt" which will contain a problem definition, and should write one text file named "output.txt" which will contain your solution (see below for format details).
- assume unix text files (single LF at end of line, no CR).
- write your own code, files will be compared automatically.

FORMAT for input.txt:

Your program (no command-line argument) should read a file 'input.txt' that defines an instance of the problem, and write out a file 'output.txt' that describes your solution. The first line of input.txt will contain two decimal numbers, the width X and the height Y of the house, plus a letter, 'B' for breadth-first or 'D' for depth-first. The second line represents the starting x,y position of the robot. The third line represents the x,y position of the charging machine. Note from the above diagram how the x,y coordinates of the top-left corner of the house are 0,0. The remaining lines represent the contents of the house, where a 0 means an empty/passable box, 1 means an impassable box, and 2 means a messy box that needs to be cleaned and is passable.

The above diagram would be represented with the following input.txt file:

```
5 5 B
1 1
3 4
111111
100101
110001
100001
120021
111111
```

There are no arbitrary limits on the X and Y of the house (it could be 5x5, or 5641x2372). However, we will not penalize you if we try to run your program on a huge house but run out of memory. We will make sure we have plenty of memory and a very fast CPU, though. Finally note that the inner topology for the house is arbitrary. For example, a house that has the rectangular outer wall, an empty spot for the robot's starting location, an empty spot for the charging station, and all ones inside is perfectly possible (although there is no solution).

FORMAT for output.txt:

Your program needs to find a path for the robot to clean the house and then go to the charging station. Your solution path need not be the shortest.

Your program should write out a file output.txt which shows the same house map as was in input.txt, but with the addition of your solution path, shown by replacing every 0 or 2 along the path by an A if the robot went through the box once, B for twice, C for 3 times, etc (after Z use a … z and beyond that it's okay to crash). In addition, one last line in your file will indicate the total path length for your solution path.

Example output.txt file:

```
111111
1AA101
11A001
10A001
1ABBA1
111111
9
```

If there is no solution, your output should be the original house map as in input.txt and a path length of -1.

Finally note that we assume that you cannot move diagonally. So, an output.txt like that:

```
111A11
11A111
111A11
3
```

is a mistake. You should return a no-solution output.txt instead.
Finally remember that all messes should be cleaned, and that you can move past/over the charging station while still cleaning the apartment.

e) [10%] Which algorithm performs best in your testing? Please professionally justify your answer (e.g., a table showing statistics over many runs). You may want to think about exactly what we mean by "performs best" here, and whether there are several meanings that you want to tell us about.