CSC421 Assignment 1. Spring 2006 (10pts)

The computer programmer is a creator of universes for which he alone is the lawgiver .... No playwright, no stage director, no emperor, however powerful, has ever exercised such absolute authority to arrange a stage or a field of battle and to command such unswervingly dutiful actors or troops.

Joseph Weizenbaum

`Student Name:` `Student Number:`

`Instructor George Tzanetakis`

Question	Value	Mark
1	2	
2	2	
3	2	
4	2	
Report	2	
Total	10	

1 Overview

The goal of this assignment is to familiarize you with uninformed and informed search algorithms and their implementation. Be aware that some of the questions require much more work than others. Your deliverable will be a report describing your formulation of the problem as a search problem, experimental results and statistics of running various algorithms on random instances of the problem and the code you implemented in order to conduct the experiments. Don't hesitate to contact the instructor via email or the anonymous comment form with any questions/clarifications you might need.

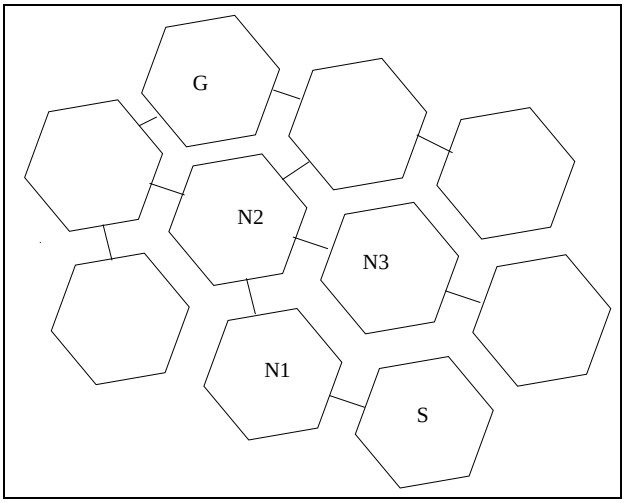

Figure 1: Hexagonal grid of locations with obstructed arcs

2 Description

Suppose a robot is searching for a path from one location to another in a hexagonal grid of locations. Figure 1 depicts such a grid. Each location has 6 neighbors but some of the connections between locations are obstructed thereby preventing passage. In the figure unobstracted connections are indicated by lines and obstructions by their absense. For example if the robot is on node N1 then it can go to node N2 or node S but NOT to node N3. The robot can move across any unobstructed connection.

3 Questions

- Formulate this problem as a search problem. Be precise about start state, goal, state-space, succesor-function etc. Write a program that generates random instances of the problem parametrized by number of locations and number of obstacles **(2pt**) .

- Implement the 3 main uninformed search strategies (Breadth-First Search, Depth-First Search and Iterative Deepening Search) for this problem. Report statistics based on solving 100 random instances of the problem for the following 4 configurations (10 locations, 20% of free obstacles), (10 locations, 50% of free obstacles), (20 locations, 20% of free ob-

stacles), (20 locations, 50% of free obstacles). The sides of hexagons that are on the boundary and have missing neighbors are considered obstacles. Depending on the grid topology you choose, you will fix the boundary obstacles and randomize the remaining. For example figure 1 contains 22 boundary obstacles and 9 inner obstacles so it is an instance of the (10 locations, 50% obstacle configuration becuase we have 19 inner connections and therefore 9.5 obstacles which can be truncated to 9).

More specifically for each configuration you will generate 100 random instances (where the start, goal, and obstructed connections are randomized) and report the following statistics: average space complexity (maximum number of nodes generated), average time complexity (number of nodes visited), actual running time (in seconds), and number of problems solved. Your implentation should be parametric and NOT hardwire any of these parameters **(2pt)**.

- Invent two admissable heuristic functions for this problem and explain why they are admissable (hint: think about relaxed versions of the same problem). Implement greedy best-first search and A* search and report the same statistics as the previous questions (4 configurations, 100 random instances etc) **(2pt)**.

- Formulate the problem as hill climbing search. Implement the basic hill-climbing algorithm for the problem and report the same statistics as the previous questions **(1pt)**.

4 Deliverable

Your deliverable will be in the form of a report. The report will describe your formulation, implementation and experimental results. For the implementation you can use any programming language you like. The book provides source code in Java, Python, and LISP that you might find useful for this assignment but you are not required to use it.

Your report should have the following sections:

- The assignment description Please include a hardcopy of this 4-page description on the front of your report.

- Introduction In this section you will formulate the problem, describe the two heuristics you propose and explain why they are admissable.
- Experimental Results In this section you will present the results of your experiments. The results should be presented in both table and bar graph form. In addition, you should try to comment and interpret your experimental results based on your knowledge of the various search algorithms. For example, do they agree with your expectations.
- Implementation Overview In this section you will give an overview of your implementation choices (Programming Languages, structure of your code, conventions, data structures used etc) and any other information you think is relevant for understanding your code.
- Code listings You need to provide full source code listings of all the algorithms you implemented. If you based your code on the code provided by the book, you must mention it and still include the relevant code in your report. Your code should be written so that it can be read and understood even by programmers not familiar with the language you are using.

4.1 Grading

The questions are worth a total of 8 points. The remaining 2 points will be based on the quality of the report.

4.2 Extra Credit (2pt)

Implement a user interface for visualizing the random instances, showing paths and running experiments as well as visualizing the execution of the different algorithms.