

UNIVERSITY OF WATERLOO  
School of Computer Science

CS486/686

Introduction to Artificial Intelligence

Spring 2011

ASSIGNMENT 1

(Due: Thursday, May 26, 2011 at start of lecture)

**Purpose:** This assignment is intended to further your understanding of problem-solving by search

1. (40 points)

You are to implement a solution to the Wordsnakes problem (described in the class handout) to solve a word puzzle problem. You may solve the problem simply, or you may try out different heuristics to improve the solution for extra marks.

You have been given starter code in the `wordsnake.zip` file in the Assignment 1 folder within “Assignments” in the ProblemSolving folder under Lessons on our UW-ACE website. This file also contains some useful example files.

The Wordsnake problem can be represented as a search on a graph, where each node in the graph is a word that is to be included in the wordsnake. You are given utility classes for data structures to represent the graph. Your main task is to implement a `Graph.java` class to search through the graph of words to build the wordsnake with the longest score. Recall that the number of overlapping letters in adjacent words is squared then added to the overall score for the wordsnake.

Although you have been given the starter code, feel free to use any representation to encode a graph data structure. You may also use any programming language to implement your problem.

In this puzzle, we are interested in maximizing the score of the wordsnake. However, we are also interested in the cleverness of your algorithm, and whether you can come up with good heuristics for finding the best solution more quickly. (We will not however grade you on the complexity of your algorithm although

you may wish to include an informal description of its complexity in your solution.)

(b) (20 pts)

Run your algorithm on the `words.txt` file included in the starter code. For full marks, test out your algorithm on three or four other sets of words that demonstrate the versatility of your algorithm. For example, you can try out word sets with minimal overlaps, maximal overlaps, lots of conflicts, etc. Be creative! There is no right or wrong answer in this assignment. We want you to try things out and experiment.

(c) Bonus marks of up to 20 points will be given for elegant algorithms or clever heuristics.

(d) What is in the starter code:

1. `Wordsnake.java` (the driver program)
2. `Connection.java` (supporting data structure)
3. `Node.java` (supporting data structure)
4. `words.txt` (testing input word list)

You are to write `Graph.java` (will contain the main algorithm).

(e) What to hand in:

1. A print-out of your `Graph.java` code.
2. Test run with the `words.txt` file.
3. Additional test runs with a listing of the word set used in each run. No more than 5 additional tests is necessary for a perfect grade.
4. A description of your algorithm with highlighting of its features and “clevernesses”.
5. **IMPORTANT!** If you use any other sample code or algorithms, you must document the source and you must also add enhancements to get any marks for your work. If in doubt, ask the Instructor.