

## Midterm

### Algorithms II

---

#### Problem 1:

In the game of *word ladders*, we are given a starting word and a target word, and asked to transform the starting word into the target word by changing one letter at a time, each time creating a new, legal word. The sequence of words formed in this way is the “ladder”.

For instance, given the starting word **foe** and the target word **hat**, we might find the following ladder:

```
foe
fox
fax
fat
hat
```

Your task is to write a program that tries to solve the word ladder game. You will be given a list of about 1,000 three letter words that are legal “moves” in the game. Your program should allow the user to type in a starting word and a target word.

If no word ladder can be found between the two words, then the program should report this. Otherwise, it should print out the word ladder that it found to connect the two words.

Your list of three letter words is available on Blackboard as *wordlist\_threes.txt*, or on my webpage as [http://people.sc.fsu.edu/~jburkardt/datasets/words/wordlist\\_threes.txt](http://people.sc.fsu.edu/~jburkardt/datasets/words/wordlist_threes.txt)

A) Describe your approach to this problem in words. A paragraph or two will be enough. In particular:

1. Explain what this problem has to do with graph algorithms;
2. Describe the algorithm you will implement;
3. Explain how you dealt with the special features of this particular problem in order to implement the algorithm;

B) Demonstrate your program on the following pairs of starting and target words by displaying the output from your program:

1. from **yes** to **nay**;
  2. from **fox** to **gnu**;
  3. from **oca** to **cub**;
-