(364-1-1441) Foundations of Artificial Intelligence

Problem Set 2: Sorry Seems to be the Hardest Word?[1] Not in this case...

Due: 1/1/2025

You need to submit both code and written answers. Problem 1 is a programmatic one, but there are things there you need to submit in writing. Problems 2 and 3 only require written answers and not programming. You will submit one `q1.py` file containing your code, and one `answers.pdf` file containing your written, typewritten (not a scan of handwriting) answers, in English or Hebrew.

Make sure your code compiles, and the output matches what is requested. Your grader will not debug your code, and if it does not compile or output correctly, they will not be in charge of fixing your errors, even if their cause was a very minor mistake.

Also, to simplify your work process, as well as the grader's, please name your variables and methods in a meaningful way (e.g., name a function `heuristic_calculation` and not `my_awesome_function`).

1 Problem 1: Word Search II

Note: Problem description identical to previous problem set

In this exercise you will build a function that checks for two words given to you if there is a chain of real words (from a dictionary given to you) such that the beginning word is one of those given to you, the final word is the other given to you, and in between, only one letter has changed from one word to another (including adding or removing a letter). So, for example, if you receive the words "late" and "kite", one option would be:

```
late
lite
kite
```

Another option would be:

```
late
lite
lit
kit
kite
```

Adding a word to the chain by adding or removing a letter costs 1/4 when the changed letter is a vowel (one of the letters {`a,e,i,o,u`}), and costs 1 when the letter is a consonant (any letter that is not a vowel). If adding a word to the chain

[1] https://www.youtube.com/watch?v=c3nScN89Klo

by changing a letter, the price is according to the letter that is now in the word (i.e., if a letter has changed to a vowel the price is $1/4$, and if the a letter changed to a consonant the price is is 1). We prefer less costly word paths.

In this exercise you will only implement A* algorithm to solve this. However, additional algorithms will be coming, so keep that in mind. . .

You will write a function in python called `find_word_path`:

```
def find_word_path(starting_word,goal_word,search_method,detail_output):
```

The function takes 4 variables (in this order of input):

`starting_word` This is the beginning of your search. You can assume this is a word that is in the dictionary (though you can write a function to check this, which will probably help in your debugging).

`goal_word` word you wish to reach from the starting word via the process. You can assume this is a word that is in the dictionary (again, a function checking its legality will probably be helpful to you)

`search_method` This will be an integer. Valid values are:

1. **This is just from the previous exercise, it won't be checked for correctness.** An A*-heuristic search. You choose the heuristic. It cannot be trivial (i.e., all 0).
2. A hill climbing algorithm. It should be restarted 5 times (if it didn't find the answer), using adjacent words to the original starting point.
3. Simulated annealing. **You choose the temperature**. In your submitted answers, containing the answers to the questions, you will detail your temperature schedule. It should not go longer than t=100.
4. A local beam search, with the number of beams (k) being 3.
5. A genetic algorithm. Population size is 10. In your submitted answers, containing the answers to the questions, explain the idea of your genetic algorithm (how are you representing solutions, how do you assess solution quality, how do you combine solutions and what is your mutation).

`detail_output` This is a binary variable. When it is false, e.g.,
`find_word_path("late","kite",2,false)`
your output is like the text above – you give the full chain of changes of words, with a single word on each line. The first one is the *starting word*, and every next line contains a word that is in the dictionary file (see below) and different in one letter from the line before it, until the last line is the *goal word*. If no path was found, the output is `No path found`.

If the binary variable is true, for the **first transformation** (from the starting word to the second word) you need to print out your work process, so for search method:

A*-heuristic search Print out the heuristic value of the word you just chose (so the one following the initial starting position). So for the call
`find_word_path("late","kite",1,true)`
your printout can possibly be

```
late
lite
Heuristic: 2.75
lit
kit
kite
```

With your heuristic value replacing, of course, the example 2.75 value given here.

Hill climbing No change in output.

Simulated annealing Instead of the heuristic, show every action you consider (in the first step following the initial word) and if the probability of taking it.

Local beam Instead of the heuristic, show your “bag” of actions considered at the first stage, and which actions did you choose.

Genetic algorithm Instead of the heuristic, print out the new population after the first step.

In order to know what is a valid word, you can use the file `dictionary.txt` downloadable from the course website. It contains a single valid word per line. **Please ignore capitalization**, so each word is legal if when you convert it to all lowercase letters.
You can assume it will be in the same folder as your code file, so no need to submit it.

A particularly creative or good implementation may get a small bonus, particularly with the choice of simulated annealing temperature and genetic algorithm implementation.

2 Problem 2: CSP I

The 4-queen problem involves 4 queens on a 4x4 board, which should not threaten each other under chess rules. Formulate the problem as a CSP, and draw its constraints graph.

3 Problem 3: CSP II

What is the time complexity of the AC-3 algorithm we saw in class? Explain it fully, including your choice of the parameters you use to express the complexity.
(The course book has the answer for this, but you need to explain it in your own words)