CSE 4081: Introduction to Analysis of Algorithms Spring 2024
Final Exam
Due Date: Friday, April 26; 23:00 hrs
Total Points: 25

Honor Code: The following document is an individual exam. As such, this exam should be solved individually, without any discussion, exchange of electronic or hard documents, or any other malpractice that constitutes to cheating in an academic institution. Even a pleasant conversation that involves revealing how much of the test one completes constitutes a violation. More information about the rules and guidelines governing student conduct and academic integrity can be found in the Student Handbook. Please also consult with your instructor if you have further questions or concerns. By continuing to participate in this exam, you implicitly agree to uphold the honor code.

Please write well-tested programs in Python3 to solve the following problems. You are allowed to use any code that is posted on Canvas, or code that you have written for any of the homework problems. You may use language libraries for the many data structures learned in class. See Section 3.3 for what is allowed. Feel free to consult the appropriate language documentations for their uses. However, you are not allowed to search for specific solutions in the Internet.

Good Luck!

1 Fast Driving

Alice Luce likes cars, and she likes to drive them very fast. She is planning to visit her grandmother. Alice is in city 1, and her grandmother lives in city n. There are n cities and m roads, each road connecting a pair of cities. Each road has a speed limit, and Alice, being a good and lawful citizen, will not travel on any road more than the indicated speed limit. However, she wants to use her brand new flashy car to the fullest, as well as use roads in which she can drive very fast. In this problem, we would like to find a path from city 1 to city n in which Alice would not have to slow down much. Help find Alice a path that maximizes her slowest speed during any part of her journey from city 1 to city n.

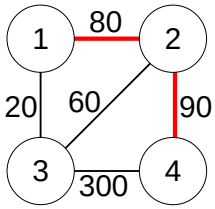

Figure 1 Graph from sample input. Optimal path shown in red.

As an example, consider the graph in the figure above. Among all paths that we can take from city 1 to city n, the path shown in red is optimal, since the slowest speed on that path is 80. On other paths, she would have to slow down to less than 80.

The input contains two integers n and m on the first line. All cities are numbered 1 through n. Then m lines follow that each describe a road using three integers x, y and v, where there is a road from city x to city y with a speed limit of v. All roads are bi-directional with the same speed limits. As output, please print a single integer that is the slowest speed on an optimal path from city 1 to city n. A sample input and output is shown below.

```
Sample Input (see fast-driving.in1):

4 5
1 2 80
3 1 20
2 3 60
4 3 300
2 4 90

Sample Output (see fast-driving.out1):

80
```

Figure 1 gives the graph of the sample input. Your program should work for $n \leq 50000$ and $m \leq 10^6$ for full credit.

2 Word Ladder

Consider a word ladder that can be formed between two 4-letter words.

lose $\rightarrow$ **lost** $\rightarrow$ **lest** $\rightarrow$ **best** $\rightarrow$ **beat**

Each word is formed from the previous word by changing exactly one letter. In this problem, we would like to find the shortest possible set of words that transforms the start word into the end word. Note that each of the words in the word ladder must be a valid word. All valid words are given in the dictionary (file `dictionary4`) and valid letters are only small case a through z. (So you can ignore words with capital letters, accents, and other punctuation marks.) You may hardcode this file into your program, but other inputs are taken from standard input.

The input contains only two words, one on each line. The first word is the start word, and the second word is the end word. As output, please print a delighted message "Yay! A word ladder is possible.", if a word ladder can be formed between the start and the end words, or "Duh! Impossible.", if no word ladder can be formed. If it is possible to form a word ladder, please print a transcript of the words in order from the start word to the end word. Note that we need a word ladder with the shortest number of transformations. If there are multiple optimal solutions, you may print any one of them.

```
Sample Input 1 (see word-ladder.in1):

lose
beat

Sample Output 1 (see word-ladder.out1):

Yay! A word ladder is possible.
lose
lost
lest
best
beat

Sample Input 2 (see word-ladder.in2):
drug
high

Sample Output 2 (see word-ladder.out2):

Duh! Impossible.
```

3 Postamble

3.1 Points Breakup

The points breakup for each problem is shown below. Please name your programs as given in the table below.

Problem	Points	Filename
Fast Driving	13 points	`fast-driving.py`
Word Ladder	12 points	`word-ladder.py`

3.2 Submission

Please submit all the files as a single zip archive `final.zip` through Canvas. The zip archive should only contain the individual files of the assignment, and these files should not be inside folders.

```
$ zip final.zip <space separated list of files>
```

This will make sure that you don't put files into unnecessary folders. Moreover, please do not include other extraneous files, including pre-compiled class or object files. We will compile your solutions before executing them so these files are not necessary.

3.3 Proper Use of Libraries

For this assignment, you are allowed to use the following data structure utility libraries in Python3. In general, you are allowed to use libraries for input and output, math and random functions.

Python	tuples, lists, sets and dicts as defined in the Python language. Also allowed are `deque` and `heapq` from `collections`.

You may use the following `Geeks for Geeks` resources for more information, or please check out the official language documentations.

Python: `https://www.geeksforgeeks.org/python-programming-language/`