University of Arizona	Homework#3	tf-idf Scoring
CSC 4/583 Text Retrieval and Web Search	Due date:	April 8, 2025
Spring 2025	**Name:**	
Instructor: Eduardo Blanco		

Instructions

- Push your code on GitHub and modify the README file with your name so we can map your GitHub user name to your official name in d2l (programming part). Submit your report to gradescope (written part).
- Steps to get started:

```
# Accept the invite link for the assignment.
#   Doing so will create a repo named ASSIGNMENT_NAME-YOUR-GITHUB_USER
#
# First, include a README file with your name so we can map your GitHub user name
#   to your official name in d2l
#
# Then, set up the environment and get started:
> git clone YOUR_REPO
> cd YOUR_REPO
> python3 -m venv .env
> source .env/bin/activate
> pip install -r requirements.txt
```

- You can play with the search engine using the following command: `python3 boolean_engine.py wiki-small.txt`. For example:

```
> python tfidf_engine.py wiki-small.txt
Query: information
[2025, 963, 4023, 214, 4243, 3598, 4809, 4162, 2453, 2427]
Query: the
[3776, 2541, 4205, 3526, 5566, 532, 4231, 4111, 1742, 3617]
Query: cool
[673, 3989, 2853, 5155, 0, 1, 2, 3, 4, 5]
Query: the cool
[673, 3989, 2853, 5155, 3776, 2541, 4205, 3526, 5566, 532]
```

- Grading of the programming part will be primarily via test cases, but we will check that you implement the tf-idf searh engine the way you are supposed to. Answers to written questions will be based primarily on your justifications. You need a working implementation to answer the written questions.

  You have access to most test cases. You can run the test cases locally on your computer with the following command: `> pytest`. For details about which test cases pass and fail, use `> pytest --verbose`.

- HW#3 is worth 70 points and 30 points of your final grade (programming and written part) but graded out of 100 points (undergraduate students) or 150 points (graduate students).
  - Undergraduate students do not need to submit the questions marked as "graduate students," but must submit all other questions (or be ok loosing points). If they submit answers to questions marked as "graduate students," their grade will be capped at 100 points. See HW#1a for an example.
  - Graduate students must do questions marked as "graduate students" or be ok loosing points for their final grade. See HW#1a for an example.

Question 1 [70pt]
TF-IDF scoring (programming). We are going to modify the search engine to return the top 10 ranked documents for a query. Queries will be a list of terms, and the answer will always the top 10 ranked documents for the query.

You should use *lnc.ltn* tf-idf weighting, that is, *lnc* for documents and *ltn for queries*. Check the example enclosed for a small example. Your implementation is most likely working if you pass the test cases.

Question 2 [30pt]
TF-IDF scoring (written). Answer the following questions:

1. What is the normalized weight of term *lincolnshire* in document 2478?
2. What is the document frequency of term *lincolnshire*?
3. What is the normalized weight of term *the* in document 1098?
4. What is the document frequency of term *the*?
5. List the 20 terms with the highest normalized weight in all documents (term, document, normalized weight). What can you say about these terms?
6. List the 20 terms with the highest document frequency (term, document frequency). Do they look like stop words?
7. Does any document contain the terms *racehorse* and *regulations*?
8. How many documents contain the term *racehorse* or *regulations*?
9. What documents are retrieved for query *racehorse regulations*? Can you justify why those documents and in that order?
10. What documents do you get for terms that are not present in the collection? Why?

Question 3 [50pt]
TF-IDF scoring (written). This question is for graduate students.

In class the instructor stated a version of "lnc.ltn is the way to implement tf-idf because it produces better rankings." Well, it is time for you to show with some examples that this is the case.

Implement tf-idf with "lnc.lnn" and compare the results of the queries in the test cases (wrt lnc.ltn). Can you justify which ranking is better?

You do not need to submit your code. You do have to do a qualitative analysis: read the ranked documents returned with both versions of tf-idf, and discuss (based on your intuition and human judgment) which ranking is better. You cannot prove anything, but you can certainly explain (with a bunch of sentences for each change in the rankings) why you think that one ranking is better than the other (for each query).