CSC 483/583: Assignment #2 (75 pts)

Due by 11:59 P.M., October 4
(upload to D2L or, if on paper, turn it in in class)

Because the credit for graduate students adds to more than 75 points, graduate students grades will be normalized at the end to be out of 75. For example, if a graduate student obtains 80 points on this assignment, her final grade will be $80 \times 75/95 = 63.2$. Undergraduate students do not have to solve the problems marked grad students only. If they do, their grades will not be normalized but will be capped at 75. For example, if an undergraduate student obtains 80 points on this project (by getting some credit on the grad students only problem), her final grade will be 75.

Note that the first six problems do not require coding. Only problem 7 requires coding.

Problem 1 (5 points)

Suggest what normalized form should be used for these words (including the word itself as a possibility). Justify your decision.

- 'Cos
- Shi'ite
- cont'd
- Hawai'i
- O'Rourke
- ain't

Problem 2 (5 points)

Assume a biword index. Give an example of a document (could be a made up paragraph) which will be returned for a query of "New York University" but is actually a false positive which should not be returned.

Problem 3 (15 points)

Shown below is a portion of a positional index in the format: term: doc1: ⟨position1, position2, ...⟩; doc2: ⟨position1, position2, ...⟩; etc.

angels: 2: ⟨36,174,252,651⟩; 4: ⟨12,22,102,432⟩; 7: ⟨17⟩;

fools: 2: ⟨1,17,74,222⟩; 4: ⟨8,78,108,458⟩; 7: ⟨3,13,23,193⟩;

fear: 2: ⟨87,704,722,901⟩; 4: ⟨13,43,113,433⟩; 7: ⟨18,328,528⟩;

in: 2: ⟨3,37,76,444,851⟩; 4: ⟨10,20,110,470,500⟩; 7: ⟨5,15,25,195⟩;

rush: 2: ⟨2,66,194,321,702⟩; 4: ⟨9,69,149,429,569⟩; 7: ⟨4,14,404⟩;

to: 2: ⟨47,86,234,999⟩; 4: ⟨14,24,774,944⟩; 7: ⟨199,319,599,709⟩;

tread: 2: ⟨57,94,333⟩; 4: ⟨15,35,155⟩; 7: ⟨20,320⟩;

where: 2: ⟨67,124,393,1001⟩; 4: ⟨11,41,101,421,431⟩; 7: ⟨16,36,736⟩;

Which document(s) if any match each of the following queries, where each expression within quotes is a phrase query?

1. "fools rush in"
2. "fools rush in" AND "angels fear to tread"

Problem 4 (5 points)

Write down the entries in the permuterm index dictionary that are generated by the term "mama".

Problem 5 (20 points)

Compute the edit distance between "paris" and "alice". Write down the 5×5 array of distances between all prefixes as computed by the edit distance algorithm in Figure 3.5 in

IIR. For each cell in the matrix, use the four-number representation to keep track of your intermediate results.

Problem 6 (10 points) GRAD STUDENTS ONLY

Consider the following fragment of a positional index with the format:

word: document: ⟨position, position, ...⟩; document: ⟨position, ...⟩
...

Gates: 1: ⟨3⟩; 2: ⟨6⟩; 3: ⟨2,17⟩; 4: ⟨1⟩;
IBM: 4: ⟨3⟩; 7: ⟨14⟩;
Microsoft: 1: ⟨1⟩; 2: ⟨1,21⟩; 3: ⟨3⟩; 5: ⟨16,22,51⟩;

The $/k$ operator, word1 $/k$ word2 finds occurrences of word1 within k words of word2 (on either side), where k is a positive integer argument. Thus $k = 1$ demands that word1 be adjacent to word2.

1. Describe the set of documents that satisfy the query Gates /2 Microsoft.
2. Describe each set of values for k for which the query Gates $/k$ Microsoft returns a different set of documents as the answer.

Problem 7 (25 points undergrads, 35 grads)

Implement an inverted index that supports proximity search. Your program must take in one file containing one document per line, in a format similar to the one from Assignment #1 (see that assignment for a detailed description of the format). For example, you can use the file below to test your code:

```
Doc1 breakthrough drug for schizophrenia
Doc2 new schizophrenia drug
Doc3 new drug for treatment of schizophrenia
Doc4 new hopes for schizophrenia patients
```

To code this problem, you can use any programming language that is familiar to the instructor (C/C++, Java, Scala, Clojure, Python). You can use data structures available in your programming language of choice, e.g., dictionaries in Python or hash maps in Java/Scala, but you are **not** allowed to use open-source code that implements inverted indices, such as Lucene. You have to implement the inverted index and corresponding search operations from scratch.

The code submitted **must** compile and run. You **must** also include in your submission a `Makefile/pom.xml/build.sbt` file or shell script that allows the instructor to run the code, together with a `README` file that describes usage.

Please implement the following:

1. **(25 points)** Construct a positional index and add support for Boolean proximity queries using the $/k$ operator. That is, word1 $/k$ word2 finds occurrences of word1 within k words of word2 (on either side), where k is a positive integer argument. Hint: use the algorithm from Figure 2.12 in the IIR textbook. What does your code return for the file above and the query: schizophrenia /2 drug? How about schizophrenia /4 drug?
2. **(GRAD STUDENTS ONLY: 10 points)** Modify the above algorithm to be directional. That is, the query word1 $/k$ word2 must return occurrences of word1 strictly **before** word2, within k words. What does your code return for the file above and the query: schizophrenia /2 drug?