

Goal

Create a search engine. You should build an **inverted index** and **ranking function**. You could spend way more time than we allot you for this task. Please make the right tradeoffs to finish the task on time. You should expect to be done with the following at the end of the code test.

- Have a working search engine. I should be able to enter a query and get back results.
 - Should support AND queries. This means ALL the search terms are in the document.
 - Should support OR queries. This means at least one of the search terms is in the document.
- Have a simple ranking function to sort the output
- Have working code ready for review

Expectations:

Due to the limited time you shouldn't focus on making a fancy UI. A command line interface to the program is acceptable. You should NOT use prepackaged search engines like Elasticsearch or Lucene. You are free to use the internet and any language you choose. Do NOT phone a friend :) You should index the files we have provided. Tf-idf is an ideal ranking function but it's hard to implement if you are not familiar with it. You should do something better than random. Do NOT implement a linear scan of the text.

Demo:

1. Review the code with the team
 - a. Performance considerations BIG-O
 - b. Tradeoffs you made
 - c. Error conditions
2. Run an ad-hoc query with AND criteria
3. Run an ad-hoc query with OR criteria

Example:

Insert the following documents into your search engine

1. My sister is coming for the holidays.
2. The holidays are a chance for family meeting.

3. Who did your sister meet?
4. It takes an hour to make fudge.
5. My sister makes awesome fudge.

Search for “who” should give you back document 3

Search for “fudge” should return documents 4 and 5. Not specifically in that order.

Reference Material:

https://en.wikipedia.org/wiki/Inverted_index

<https://www.quora.com/What-is-an-intuitive-description-of-how-Lucene-works>

<http://www.tfidf.com/>

http://www.site.uottawa.ca/~diana/csi4107/cosine_tf_idf_example.pdf

https://en.wikipedia.org/wiki/Jaccard_index