To build a search engine, we first need to crawl a certain number of pages by BFS.

- To compute the pagerank of each page. We first gives a unique id for each page. Then we build a adjacency matrix to represent the directed from sources pages to target pages. And this will be our input.
- By reading the input, we initialize the pagerank by 1/n, and n is the number of page. And for each iteration we update the pagerank of each page basing on the information of its target pages. After a certain number of iteration the pagerank value will converge and the output will be a numeric value for each page which indicates the popularity of that page.

Calculating the pagerank is just a start for building a search engine. As a search engine, it needs to find pages that are related to the query words and help users to locate information for the query words. Thus we need to analyze the content of each page.

- The input for project 4 is the content of each page. We use a table which contains the URI of the page and the corresponding content of the page as input. And for each URI there is an unique document id associated with it.
- Then, for each page document, we compute the frequency of each word in the document, and produce a table to record the word, frequency and document id.

Basing on the inverted indexing table and pagerank table we should be able to filter some useful pages by given some query words.

- The input for project are index table, data table, and pagerank table. Index table provides the frequencies of query word in each document. Data table contains the URI for each document, and pagerank table records the pagerank for each document.
- With invert indexing table, we can compute TD-IDF score easily. This score tells us the relevancies of documents responding to the query word. A high score indicates a document contains useful information for the query word. By given the document id, we can directly access the precomputed pagerank value. Then we can combine those two scores by some multiplier. In the end we rank the final score and provide the top 20 or 100 pages for the query word.

There are more things we can add to optimize the results.

- Spam detection. There are a lot of pages trying to game the search engine to get high pagerank value. Those pages repeat the keywords again and again or make their text invisible. The relevant pages would be neglected. There are many techniques to detect spam website. One of the most easiest way is to create documents with spammy content. Then for each document we crawl, we compute the TD-IDF of each word respect to those spammy documents and sum them up. If a document has a high score, then that page may contain spammy content and we should neglect it.
- We can also rely on the feedback of user. We can ask user to how much they satisfy with the result of searching. And we prioritize the pages with higher satisfaction. It is a more straightforward way to find relevant pages.