Building A Search Engine

By Sahil Mutneja
under the mentorship of Dr. Dileep A.D.

Problem Statement

- Search Engine that provides the user with a list of experts when queried against a specialization
  - The experts will be from the premium Institutes(IIT’s, NIT’s) spanning across India
- The list will be such that the most relevant of the result will be towards the top.

Final Outcome

- The result will comprise of the links of web pages of the specialist along with the top-notch work done on the queried specialization.
- After analysing the crawled content, we will be building a web graph/network wherein nodes will be the specialization and the edges will be the links of the webpages or some relevant content

Description of the project

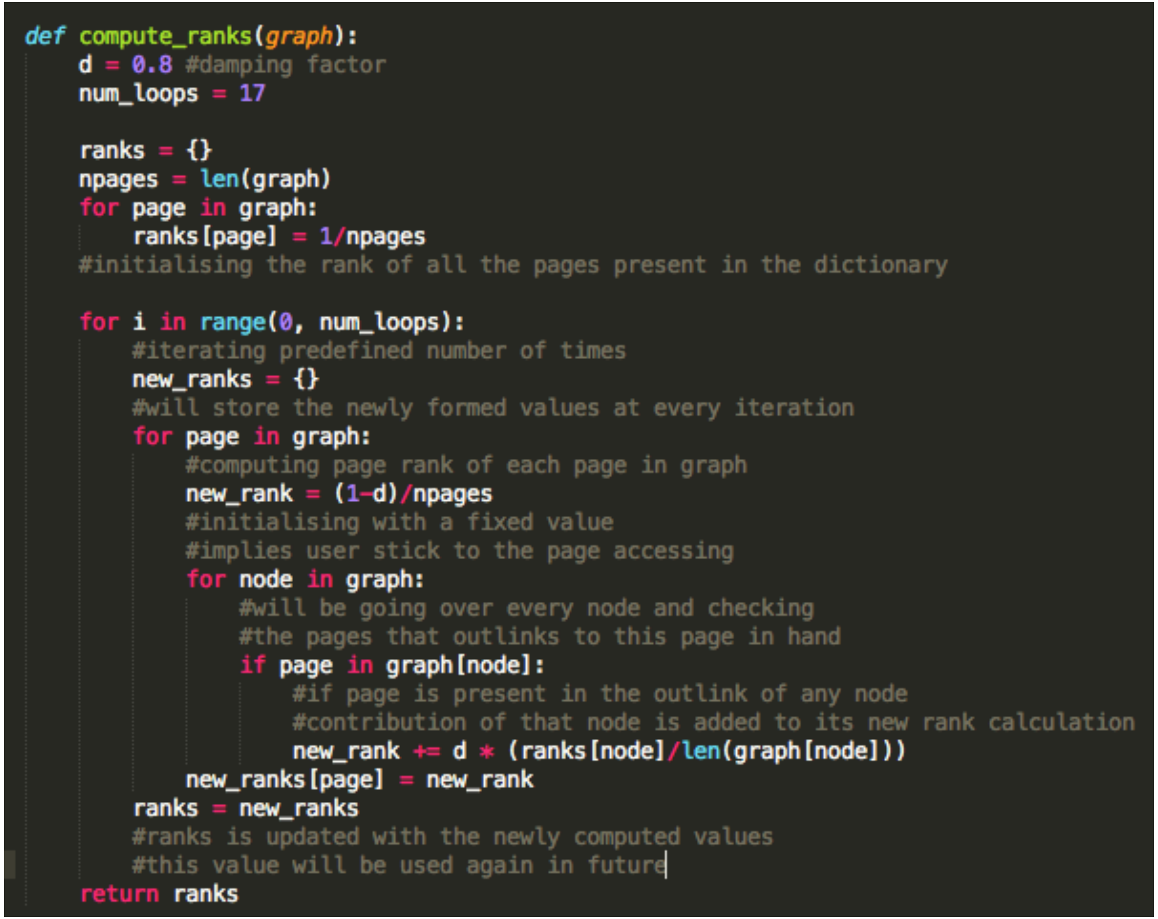
Decomposing the Problem Statement gives us three subparts:

1. Web Crawling
2. Indexing
3. Searching

Plan as per the last review meeting

Work to be Completed	Tentative Date	Current Status
Build a Crawler	11th October 2014	Built a crawler that crawls the relevant content of IIT Mandi
Formatting via Parser	7th November 2014	Formed a Indexer wherein specialization lookup is supported.
Search Optimization Techniques	7th March 2015	Dictionary based implementation of Indexer to have all the operations in O(1) time
Implementation of ranker and Searcher for IIT Mandi	15th March 2015	Terminal based implementation of Search Engine on the scale of IIT Mandi

PageRank Algorithm Implementation



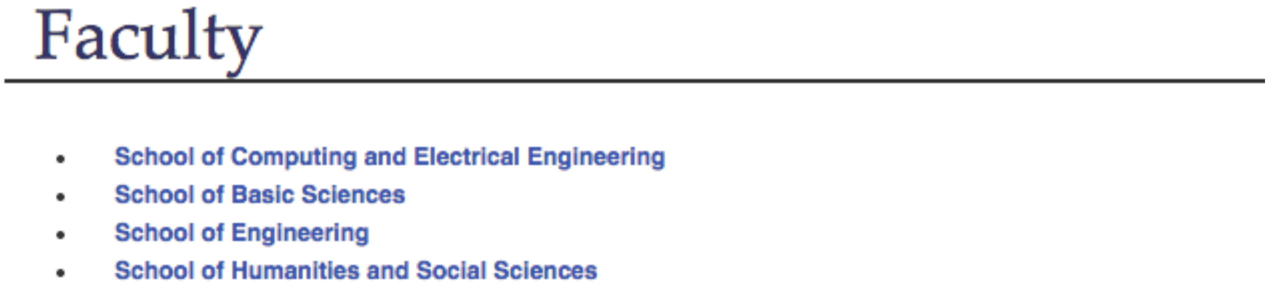
```python
def compute_ranks(graph):
    d = 0.8 #damping factor
    num_loops = 17

    ranks = {}
    npages = len(graph)
    for page in graph:
        ranks[page] = 1/npages
    #initialising the rank of all the pages present in the dictionary

    for i in range(0, num_loops):
        #iterating predefined number of times
        new_ranks = {}
        #will store the newly formed values at every iteration
        for page in graph:
            #computing page rank of each page in graph
            new_rank = (1-d)/npages
            #initialising with a fixed value
            #implies user stick to the page accessing
            for node in graph:
                #will be going over every node and checking
                #the pages that outlinks to this page in hand
                if page in graph[node]:
                    #if page is present in the outlink of any node
                    #contribution of that node is added to its new rank calculation
                    new_rank += d * (ranks[node]/len(graph[node]))
            new_ranks[page] = new_rank
        ranks = new_ranks
        #ranks is updated with the newly computed values
        #this value will be used again in future
    return ranks
```

Seed Page from which crawling begins

http://www.iitmandi.ac.in/institute/faculty.html



Faculty

- School of Computing and Electrical Engineering
- School of Basic Sciences
- School of Engineering
- School of Humanities and Social Sciences

Indexer and Ranker

- As of now, a terminal based search engine on the scale of IIT Mandi has been built.
- As the script is run, it crawls the relevant pages lying in the domain of IIT Mandi. After crawling is done, the pages are added to the Indexer dictionary and graph data structure where for each page its outlinks are recorded.
- The final Indexer content consists of a dictionary where the key is a string that represents the specialization and the value is the list of web pages.
- Whenever user enters a query, dictionary linear look up starts and all the pages are added to the final result list and then sorted based on their ranks
- For IIT Mandi, list contains 18,377 entries in the Indexer.

Things in Hand and Future Targets

- Research on the techniques that will make the Indexer more enhanced and more precise towards better results. Inverted Index, ML algorithms are some of the many.
- Improvement in the terminal based version of Search Engine.
- Transforming the terminal based script to user friendly web application.
- Expanding the crawled data to a multiple number of pages in order for the result to cover wide aspects(specific to IIT Mandi).
- Making the system scalable to a large number of machines for storing the processed indexer data.
- After ensuring the scalability, focus will be on the front end and the improvement of existing structure.

Course of Action with Time Frame (7th and 8th Sem included)

Work to be Completed	Tentative Date
Research on Text Classification, ML Algorithms for Indexer Enhancement and Algorithms for Ranker	In Winter Vacations
Search Optimization Techniques	7th March 2015
Implementation of Ranker and Searcher for IIT Mandi	15th March 2015
Extending domain of Crawler and Parser to more institutes and build Indexer	27th March 2015
Setting up of Distributed System for huge Indexer data	15th April 2015
Ranker Collaboration with large data	19th April 2015
Building Front End and Final Touches	30th April 2015