# COMP 4601A
# Fall 2022 – Assignment #1

## Objectives

The goal of this assignment is to implement a basic search engine using the concepts and tools covered in the first half of the course. To complete this assignment, you will need to implement a web crawler, a RESTful server, and a browser-based client that will allow a user to perform searches.

There may also be a bonus opportunity for taking part in the distributed search engine experiment. More information about this will be posted later.

## Submission Requirements

There will be several requirements for submission:

1. The code for your assignment must be submitted on Brightspace. You should not submit your database files. This submission must include a README file with your name, your partner's name (if applicable), a summary of the parts of the assignment that you did/didn't complete successfully, and a link to your video demonstration.
2. Your assignment must also be deployed to OpenStack so the TA can execute search requests. Include the URLs the TA should use to query your search engine in your README file.
3. You must record a ~5-15 minute (I'm not sure how long it will take, so giving a flexible range) video where you:
    a. Demonstrate your search engine's functionality for both crawled sites.
    b. Discuss the design of your implementation (see the end of the document for some ideas of things you could discuss).
    c. Discuss the personal site you selected, the challenges that it presented, and how you addressed these challenges.
4. Your video demonstration must be hosted online. You can use any tools/host you want. Carleton offers support for recording and hosting videos using Kaltura and MediaSpace. See https://carleton.ca/capture/kaltura-video-assignment/ for details. If you are using MediaSpace, you can set your video to 'Unlisted' and share the link. Ensure the video can be viewed even if you are not logged into your account.

**Partners submitting the assignment should make a single submission that contains both partners' names and student numbers in the README file.**

# Assignment Requirements

The web crawler portion of your assignment must be capable of crawling the following:

1. The fruit example site. Start at people.scs.carleton.ca/~davidmckenney/fruitgraph/N-0.html and crawl the entire site (1000 pages).
2. Another site of your choosing. Limit the total number of crawled pages (500-1000). It is suggested to limit your crawl within the same domain that you begin. You can design your selection policy to focus on any pages or resources you deem important. You are not required to crawl non-HTML resources but can choose to do so.

Your crawled data must be stored in a database for persistence. Your crawler must also perform PageRank calculations and store the values for each page in the database.

Your RESTful web server must read the data from the database, perform required indexing, and provide relevant, ranked search results for any valid request. Your server must support GET requests for at least the following endpoints:

1. /fruits – represents a request to search the data from the fruit example
2. /personal – represents a request to search the data in the alternate site you selected

Both of your search endpoints (/fruits and /personal) must support at least the following query parameters:

1. q – a string representing the search query the user has entered, which may contain multiple words
2. boost – either true or false, indicating whether each page should be boosted in the search results using its PageRank score
3. limit – a number specifying how many results the user wants returned (minimum 1, maximum 50, default 10)

The browser-based interface for searching must allow the client to specify:

1. The text for their search
2. Whether they want the results to be boosted or not using PageRank
3. The number of results they want to receive (minimum 1, maximum 50, default 10)

The search results displayed in the browser must contain:

1. The URL to the original page
2. The title of the original page
3. The PageRank of the page within your crawled network
4. A link to view the data your search engine has for this page. This must include at least the URL, title, list of incoming links to this page, list of outgoing links from this page, and word frequency information for the page (e.g., banana occurred 6 times, apple occurred 9 times, etc.). You can also display any additional data you produced during the crawl.

# Potential Discussion Points

Below are some things to consider addressing in your demonstration video. This list is not exhaustive, so include any other meaningful analysis you think is valuable.

1. How does your crawler work? What information does it extract from the page? How does it store the data? Is there any intermediary processing you perform to facilitate the later steps of the assignment?
2. Discuss the RESTful design of your server. How has your implementation incorporated the various REST principles?
3. Explain how the content score for the search is generated.
4. Discuss the PageRank calculation and how you have implemented it.
5. How have you defined your page selection policy for your crawler for your personal site?
6. Why did you select the personal site you chose? Did you run into any problems when working with this site? How did you address these problems?
7. Critique your search engine. How well does it work? How well will it scale? How do you think it could be improved?