CSE 3024: Web Mining **Slot: L51 + L52**

Online Submission Deadline: 03rd September 2020

Crawler Implementation, Index Compression, TF-IDF

[2.5 x 4]

- **Upload your code and result as a single PDF file in VTOP and MOODLE**
- **File should contain**
  - **Question**
  - **Code**
  - **Result / Output screen (including contents of all generated files)**

---

1. Write a python program to
   a) show the implementation of a concurrent depth-first crawler (No. of threads = 5 and depth = 5).
   b) Develop the crawler program to handle various challenges (such as Parsing, Stemming, Lemmitization, Link Extraction, Canonicalization, Spider Trap etc.) faced by crawler while implementing.
   c) Based on the contents retrieved, prepare one inverted index file (with proper representation).

2. Write a python program to show the implementation of Golomb Encoding-decoding technique.
   a) Encode x=25, 37, with b=11 and b=16.
   b) Decode the Golomb encoded sequence 1111111110010001101 with b = 10.

3. Write a python program to extract the contents (excluding any tags) from two websites
   https://en.wikipedia.org/wiki/Web_mining
   https://en.wikipedia.org/wiki/Data_mining

   Save the content in two separate files. Construct a trie based on the content retrieved in using HashMap / B-Tree / Dictionary. Write a program to show the implementation of **Predictive Typing** and **Auto-Correct** using the trie prepared.

4. Write a python program to extract the contents (excluding any tags) from the following five websites
   https://en.wikipedia.org/wiki/Web_mining
   https://en.wikipedia.org/wiki/Data_mining
   https://en.wikipedia.org/wiki/Artificial_intelligence
   https://en.wikipedia.org/wiki/Machine_learning
   https://en.wikipedia.org/wiki/Mining

Refined the contents by applying stopword removal and lemmatization process. Save the refined tokenized content in five separate files. Considering a vector space model and do the following operations according to the query "Mining large volume of data".

- Bag-of-Words (Document corpus)
- TF (Document corpus)
- IDF (Document corpus)
- TF-IDF (Document corpus)
- TF-IDF (Query)
- Normalized (Query)
- Normalized - TF-IDF (Document corpus)
- Cosine Similarity
- Euclidean Distance
- Document Ranking (Display Order)
- Document Similarity (Among Documents)