Database Design

url_id , id_url

Mapping Indexes :

URL ⟷ page ID

- URL → page ID

  string → integer

  int getEntry(string)

  void addEntry(string, int)

- page ID → URL

  int → string

  string getEntry(int)

  void addEntry(int, string)

word - page

Inverted Indexes

word → page ID, freq

string → HashMap<ID, freq>
int int

HMap getHMap(string)

void addFreq(string, int)

page_word

Forward Indexes

page ID → words

string → string

String

void addEntry(int, string)

String getEntry(int)

page_prop

Page Properties

pageID → { Title, last modified, content length }

int → HashMap<tag, content>
string string

void addProps(int, HMap)

Vector<String> getResult(int)

parent_child , chid_parent

Link Based Indexes

pageID(parent) ⟷ pageID(child)

int → ArrayList<IDs>
int

void addEntry(int, int)

ArrayList<int> getID(int)

url_url

Mapping Indexes :

URL → URL

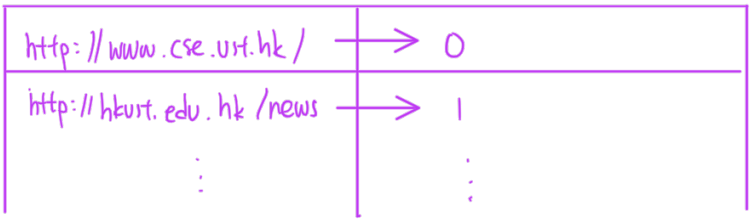
string → string

String getURL(String)

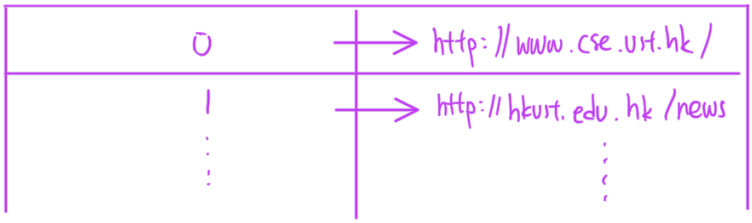
Void addEntry(string, string)

Above is the sketch of the database used in my search engine project. Every database works independently in a sense that every database does not contain any of the 8 databases as the value. Below is the explanation for each databases design:

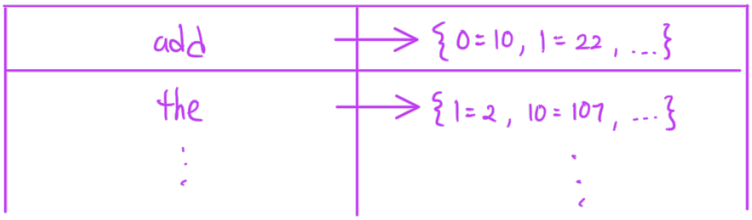
1. URL —> PageID [Mapping Index] = this database contains every URL mapped with an unique ID to it
   ex.



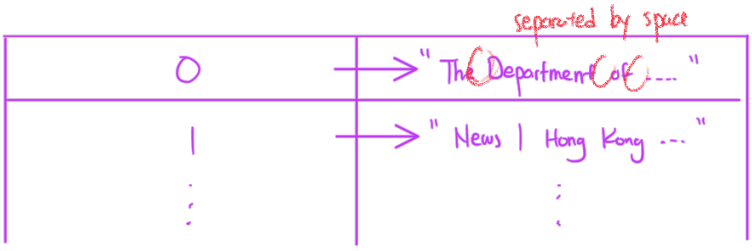
2. PageID —> URL [Mapping Index] = this database is the reversed of the first database design, which maps an ID into an URL
   ex.



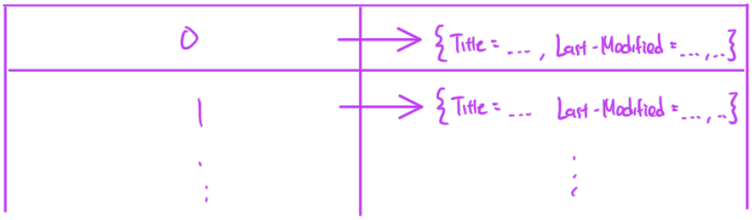
3. word —> (PageID, frequency) [Inverted Index] = this database maps every word into every pageID that has the words in it, together by how often it appears in the website as the frequency
   ex.



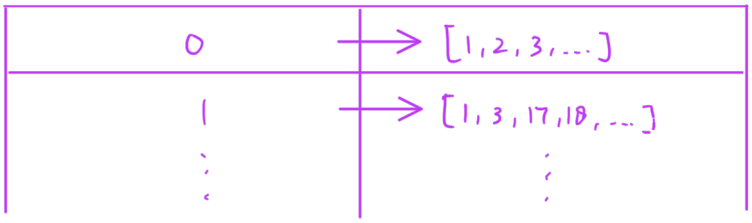
4. PageID —> words [Forward Index] = this database maps pageID to every words that reside inside the page
   ex.



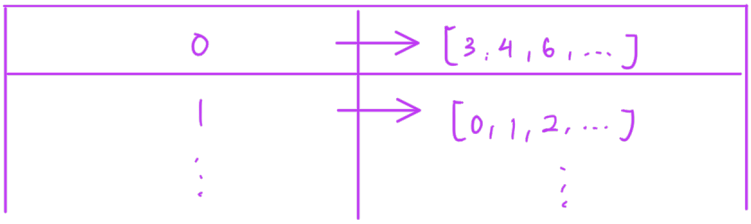
5. PageID —> page properties = this database maps pageIDs into the property of the page such as the page title, the content length, and the last modified date.
   ex.



6. Parent —> Child [Link Based Index] = this database contains every parent-child relation of every indexed urls
   ex.



7. Child —> Parent [Link Based Index] = this database contains every child-parent relation of every indexed urls
   ex.



8. URL —> URL [Mapping Index] = this database maps every URL into its corresponding actual URL. Actual URL means that if there is any redirection from the website (status code= 3xx), then the actual URL is the link inside the Location properties of a HTML response
   ex.