

Абстрактний тип даних , який необхідний для вирішення задачі дослідження - Search ADT

У цьому абстрактному типі даних програма буде працювати безпосередньо з отриманням даних через `арі`, їх збереженням та пошуком. Структура даних, на яку буде опиратися цей абстрактний тип даних - одно- та двозв'язні списки. Щоб зберігати усю інформацію разом та послідовно, у кожного знайденого файлу буде об'єкт двозв'язного списку, у якому буде зберігатися інформація про кількість переглядів, назву конференції, посилання на `ted.com` та короткий опис.

Абстрактний тип даних Search складається з таких модулів та функцій:

- `csv_reader` - `FileExplorer`:
  - `getter(i, parameters)` - знаходить у файлі за значенням `id` - і дані параметри(перегляди, посилання і т.ін.) та зберігає їх у двозв'язному списку
  - `csv_reader()` - читання та повернення файлу
  - `id_retriever()` - повернення усіх `id`
- `data_collector` - збереження, чистка файлу
- `search` - `Search(keywords)`:
  - `input_measure(keywords)` - повертає посортовані за `tf-idf` значенням список однозв'язних списків
  - `node_pusher()` - повертає отримані з `getter()` усі необхідні дані з `csv` файлу та зберігає їх у двозв'язному списку
- `normalizer`:
  - `stop_words_removal(transcript)` - видалення слів, які не містять суттєвого лексикографічного значення
  - `stemmer(word)` - виділення кореня зі слова
- `node_` - `Node(data, next)`, `TwoWayNode(data, next, previous)`:
  - `tail()` - повернення останнього елементу зв'язної структури
  - `head()` - повернення першого елементу зв'язної структури
- `flask_app`

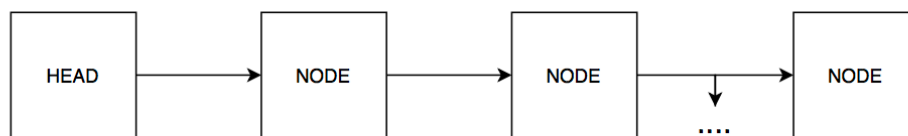
Структури даних, які використовуються у Search ADT:

- одностов'язний список
- двостов'язний список

Приблизний вигляд збереження інформації для кожної конференції:

Відео id -> назва <-> кількість переглядів <-> ted посилання <-> опис конференції

### NODE



### TWO WAY NODE

