Описание алгоритма

1. Файл загружается в память с использованием mmap
2. Из файла последовательно извлекаются слова и предложения (токенизация), в это время формируются три главные сущности программы:
    a. Обратный индекс. Отображение слова в множество пар вида <номер предложения, TF> (Важно отметить, что по построению это множество отсортировано по возрастанию номеров предложений)
    b. Таблица Term Document Frequency для каждого слова в тексте
    c. Таблица отступов. Чтобы быстро передвигаться по тексту, составляется массив, где индекс элемента - номер предложения, а значение - отступ в байтах * sizeof(wchar_t)
3. Программа ожидает поисковый запрос.

Способ выбора сниппета:

1. Входной запрос разбивается на слова и сортируется в порядке возрастания document term frequency. Из всего запроса используется не более MAX_TOKENS_TO_USE термов.
2. Собирается множество номеров предложений, в который встречаются отобранные ранее токены. Для каждого из номеров вычисляется вес предложения по формуле (i - номер отобранного из запроса терма):

$$Weight(s) = \frac{\sum_{i=1}^{n} TF(s,i) \times IDF(s,i)}{1 + |ln(ESL) - ln(length(s))|}$$

   ESL (Expected Sentence Length) - оптимальная длина предложения, которое входит в итоговый сниппет. Логарифм в знаменателе выбран для того, чтобы штрафовать длинные предложения сильнее коротких.
3. Из всех предложений выбирается не более MAX_SENTENCES_TO_USE штук, имеющих наибольший вычисленный на предыдущем шаге вес.
4. Отобранные предложения комбинируются в одну строку с разделителями и возвращаются пользователю.