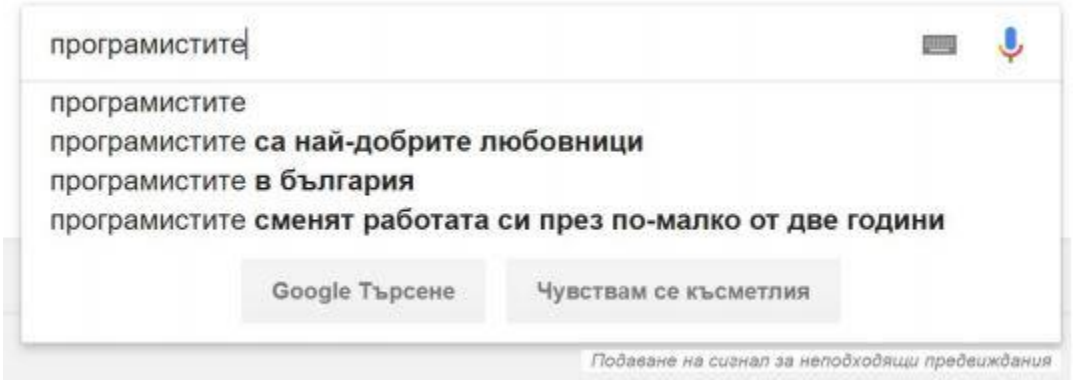
Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

Проекти

курс Структури от данни и програмиране
за специалност Информатика
зимен семестър 2018/19 г.

Autocomplete

В рамките на този проект трябва да разработите модул за autocomplete. В него ще можете сами да изберете алгоритъма, с който да работите.
Модулът трябва да предоставя функционалност подобна на тази, която позволява на Google Search да визуализира списък с предложения, когато започнете да пишете дадена фраза:



Фигура 1: Google винаги дава вярна информация

За разлика от инженерите от Google, Вие ще трябва да решите много по-проста задача. Като вход от потребителя ще получите път до текстов файл. Този файл ще съдържа N на брой символни низа. Това ще бъдат различни думи и/или фрази, по една на всеки ред. Вие трябва да заредите съдържанието на файла в подходящ контейнер. След това трябва да можете, по префикс P, съдържащ L(P) букви, да връщате списък от не повече от K предложения за това как може да се довърши префикса.

В рамките на проекта трябва да решите следните задачи:

Задача 1. Потърсете в Интернет информация за това какви подходи съществуват за решаване на тази задача. Започнете проучването си от източниците дадени в края на това описание.

Задача 2. Изберете подходяща структура от данни, с която да решите задачата. Вашето решение трябва да може да работи достатъчно бързо, така че потребителят да може да получава предложения в реално време, докато пише на клавиатурата. Също така, броят на входните символни низове N може да бъде много голям.

Помислете за това дали и какви ограничения за P да поставите. Например може да поискате L(P) да бъде поне 3 или повече букви, преди да започнете да давате предложения.

Относно K: очевидно, ако потребителят въведе префикс, който не съвпада с никой от низовете в контейнера, броят на предложенията ще бъде 0. Ако обаче има поне едно съвпадение, трябва да се помисли за това дали да се постави ограничение. Ако например има 1 000 000 съвпадения, няма как да предложите всички на потребителя. Затова потребителят трябва да може да указва някаква стойност за K, например 10, след което вашето решение трябва да предлага най-много K символни низа при дадено търсене. K трябва да може да се променя динамично по време на изпълнение. Например ако потребителят реши, че 10 низа не са му достатъчни, той може да увеличи K до 20.

Обърнете внимание и на обема памет, който ще изисква избраното от вас представяне. За целите на проекта приемаме, че входната азбука не се ограничава само до латинските букви и може да включва произволни Unicode символи. Затова и приложението Ви трябва да работи с Unicode, а не с еднобайтови char символи.

След стъпка 1 и 2, още преди да започнете да програмирате решението си, трябва да съгласувате това, което сте избрали като подход за решаване на задачата, с екипа на курса. Така ще се избегне ситуация, в която сте избрали прекалено проста или прекалено сложна за рамките на курса СД. За целта изгответе кратко описание (достатъчно е половин до една страница), в което:

- опишете накратко коя структура от данни сте избрали (може да са няколко);
- опишете кои помощни алгоритми ще използвате (ако има такива);
- кажете каква ще бъде архитектурата на решението (достатъчно е кратко описание или диаграма);
- обосновете взетите от вас решения.

Ако се затруднявате, прочетете как се правят дизайн документи, например в тази статия.

Качете описанието, например като MS-Word или PDF документ, във формата за предаване на проект и се свържете с екипа. Ще получите от нас потвърждение или забележки за неща, които трябва да се променят. Преминете към задача 3 едва след като бъдат изчистени всички забележки свързани с избрания от вас подход.

Задача 3. Реализирайте избрания от вас алгоритъм. Като минимум трябва да реализирате клас Autocomplete, който да предлага поне следните операции:

● Insert – добавя нов символен низ в контейнера;
● Suggest – получава като аргумент един символен низ – префикс, по който трябва да се направи търсете. Операцията да връща нула, един или повече символни низа – предложения за това как да се довърши префикса.
● Трябва да има възможност за извличане и промяна на стойността на K.

Задача 4. Тествайте решението си, като му подадете голям обем входни данни и го поставите под натоварване, за да проверите колко бързо то може да обработи голям брой заявки.

Задача 5. Напишете кратка програма, която демонстрира работата на решението.
Програмата трябва да може да зареди списък от символни низове от файл. След това тя трябва да позволява на потребителя да въвежда различни символни низове, да прави по тях търсене и да извежда предложенията на екрана. Програмата трябва да позволява на потребителя да променя стойността на K.

Ако имате възможност, реализирайте програмата така, че да има графичен интерфейс. Изобразете поле за въвеждане (edit box) и под или над него, в реално време, извеждайте предложенията, които решението ви генерира.

Anti Ajanki, Data structures for fast autocomplete
https://www.futurice.com/blog/data-structures-for-fast-autocomplete/

Algorithm for autocomplete?
https://stackoverflow.com/questions/2901831/algorithm-for-autocomplete