# Hacettepe University
# Department of Computer Engineering
# Assignment 2

**Subject:** Complete Matching Words via Search Key

**Submission Date:** 24.3.2017

**Due Date:** 7.4.2017 (23:59:59)

## INTRODUCTION

Binary search, as fundamental algorithm in searching, is employed to rapidly find a value in a sorted sequence. Binary search actually works on a diminishing subsequence of the starting sequence where the target value is searched which is called the search space. The whole sequence is considered the search space at the initial stage. At each stage, the median value in the search space is compared to target value and half of the search space is abolished. As a result, the algorithm leads us to have a search space consisting of a single element, the target value.
As binary searching expedite searching process, it can be used in query based searching events. Here in this homework, you will find out how sorting and binary searching can be efficiently employed for seeking specific queries entered by the client.

## AIM & DESIGN

Word completion, is a feature in which an application predicts the rest of a word a user is typing. In graphical user interfaces, users can typically press the tab key to accept a suggestion or the down arrow key to accept one of several. It speeds up human-computer interactions when it correctly predicts the word a user intends to enter after only a few characters have been typed into a text input field. Word completion works so that when the writer writes the first letter or letters of a word, the program predicts one or more possible words as choices. If the word he intends to write is included in the list he can select it, for example by using the number keys.

This experiment is designed to improve your skills on managing searching algorithms using binary search. You have to write a program that implements matching words via search keys automatically for a given set of N term. A term is a query string and an associated nonnegative weight. To perform this assignment, you will have access to a set of all possible queries and their associated weights.

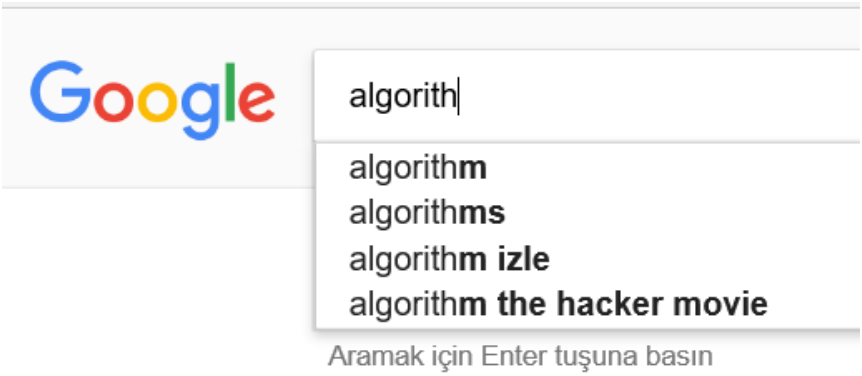


Figure 1: Sample image from the Google Search.

In this assignment,[1] you will write a program that automatically completes the matching words via search keys by sorting the terms by query string; binary searching to find all query strings that start with a given prefix; and sorting the matching terms by weight. That is, given a search key, find all queries that start with the given key, in descending order of weight.
You should implement this assignment in two stages:
a. Sort the terms alphabetically
b. Binary search to find all terms that start with the given query; sort the matching terms in descending order by weight.

**Input format.**We provide a number of sample input files for testing. Each file consists of an integer N followed by N pairs of query strings and nonnegative weights. There is one pair per line, with the weight and string separated by a tab. For example,
- The file **citiest.txt** contains over 90,000 cities, with weights equal to their populations.
-The file **wiktionary.txt** contains the 10,000 most common words in Project Gutenberg, with weights proportional to their frequencies.
- The file **alexa.txt** contains the 1,000,000 website with weights proportional to their frequencies.

**% more citiest.txt**
93827
14608512 Shanghai, China
13076300 Buenos Aires, Argentina
12691836 Mumbai, India
12294193 Mexico City, Distrito Federal, Mexico
11624219 Karachi, Pakistan
11174257 İstanbul, Turkey
10927986 Delhi, India
10444527 Manila, Philippines

[1]It is cited from an assignment at Princeton University.

10381222 Moscow, Russia
...
2 Al Khaniq, Yemen

**% more wiktionary.txt**
10000
5627187200 the
3395006400 of
2994418400 and
2595609600 to
1742063600 in
1176479700 i
1107331800 that
1007824500 was
879975500 his
...
392323 calves

Client reads the data from the file; then it repeatedly reads automatically completed queries from standard input, and prints out the top k matching terms in descending order of weight.

**Output format.**

**% cities.txt 7**

**M**
12691836 Mumbai,India
12294193 Mexico City, Distrito Federal, Mexico
10444527 Manila, Philippines
10381222 Moscow, Russia
3730206 Melbourne, Victoria, Australia
3268513 Montral, Quebec, Canada
3255944 Madrid, Spain

**% cities.txt 3**
**Sha**
14608512 Shanghai, China
1333973 Shantou, China
770000 Shangyu, China

**% wiktionary.txt 2**
**as**

703754300 as
42004400 asked
**Note.**

- The constructor should throw a java.lang.NullPointerException if query is null and a java.lang.IllegalArgumentException if weight is negative.
- Make the analysis of your program that you construct(time complexity)and in your reports, explain the reasons for using the methods you use.
- Also, be sure to test your program on many inputs.

## IMPORTANT

- The assignment must be original, INDIVIDUAL work. Downloaded or modified source codes will be considered as cheating. Also the students who share their works will be punished in the same way.
- You can ask your question via course's piazza group.
- Pay attention for the following items while coding: write English comments for your source codes, design your code according to given instructions above.

## SUBMISSIONS

You are required to submit all your code (*all your code should be written in Java*) along with a report in PDF format (should be prepared using LaTeX). The codes you will submit should be well commented. Your report should be self-contained and should contain a brief overview of the problem and the details of your implemented solution. You can include pseudocode or figures to highlight or clarify certain aspects of your solution. Finally, prepare a ZIP file named `studentID-pa2.zip` containing

- `report.pdf` (PDF file containing your report)
- `code/` (directory containing all your codes as *.Java files)

The ZIP file will be submitted via the department's submission system.You have to use "Online Experiment Submission System". http://submit.cs.hacettepe.edu.tr. Other type of submissions especially by e-mail WILL NOT BE ACCEPTED.

**LATE POLICY** It is allowed only one day late submission and it will be degraded by -10 points.

## REFERENCES

1. http://www.geoba.se/population.php?cc=world
2. https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists
3. http://www.alexa.com/topsites
4. http://http://norvig.com/ngrams/
5. https://www.ssa.gov/oact/babynames/limits.html
6. https://https://www.uspto.gov/trademark