Design and Analysis of Algorithms (CSCI 323)
Winter 2022
Programming Assignment #1
Due by class on Wed., Jan. 5, 2022

Instructions:

- Programs should generally be written in C, C++, Java, or Python. You may use any built-in functions provided by the language or its associated libraries.
- Programs should be submitted before the start of class on the date due through an online Google form that will be provided, but will generally be graded through a peer-review system. The instructor reserves the right to review any assignments and/or ask its author to explain the code, and reassign a grade as he deems appropriate.
- You are permitted to work individually or in pairs. However, even if you work in a pair, you must have your own copy of the code that runs on your own computer that you can demonstrate as needed.
- In your programming assignments for this course, you are generally permitted to utilize code found in texts or online resources provided that you
  - acknowledge the source with the text name or full URL, eithvanter at the beginning of the code or before the specific function to which it is most rele
  - understand the code and can explain it upon request
  - verify that the code in fact works
  - integrate the code into the overall format and function of the assignment
- At the top of each program source code, write the following
  - Course name and number (Analysis of Algorithms - CSCI 323)
  - Assignment Number
  - Your first and last name
  - If applicable, any students with which you collaborated (allowed, but must be acknowledged)
  - If applicable, any websites/texts from which you got code (allowed, but must be acknowledged)

Overview:

One of the most basic programming problems is "search", that is finding an item ("key") in a structure. The structure may be ordered (e.g. alphabetical or numerical order) or unordered (no particular order). There are several search algorithms available; in this assignment, we consider a list of random data and compare the following approaches:

- Linear Search
- Binary Search
- Interpolation Search
- Jump Search
- native search

The last of these is not the name of a search algorithm, rather we refer to the language's built-in search capabilities

Preliminary Tasks:

- Create a new file Assignment1. (If a configuration is required, create that too.)
- At the top of your program, place a four-line comment with the course, semester, assignment, and your name
- Write a function main() from which you will call all the functions described in Main Tasks
- Before each function, add a one line comment contain the description below as to what the function will do

Main Tasks:

[1] Write a function random_list(min, max, size, sorted) that returns a list of size random numbers in range(min, max) , and sorts it if sorted=true

[2] Write a function linear_search(list, key). See https://www.geeksforgeeks.org/linear-search/

[3] Write a function binary_search(list, key). See https://www.geeksforgeeks.org/binary-search/

[4] Write a function interpolation_search(list, key). See https://www.geeksforgeeks.org/interpolation-search/

[5] Write a function jump_search(list, key). See https://www.geeksforgeeks.org/jump-search/

[6] Add code to time each of the search functions

[7] Add code to run each search multiple times for different random keys

[8] Add code to plot the timings for the five algorithms. See https://www.geeksforgeeks.org/matplotlib-tutorial/