This program creates two arrays of random numbers that you can use to test linear and binary searches for. There is no need to provide the program with random numbers, as it will do that on its own.

The program takes two input arguments:

- n The number of randomly generated integers you want as the elements in the initial array.
- s The number of randomly generated integers you want to compare to the elements in the original array.

The program has two primary methods:

linear_search(int* A, int key, int size);

A - The initial array.

key - The value for which you are searching in the initial array.

size - The size of the initial array.

linear_search is called in the main function under a for loop that calls the function for each value in the second array of search terms.

binary_search(int* A, int key);

A - The initial array.

key - The value for which you are searching in the initial array.

binary_search only needs these two values because it is called iteratively rather than recursively.

Test cases and test results:

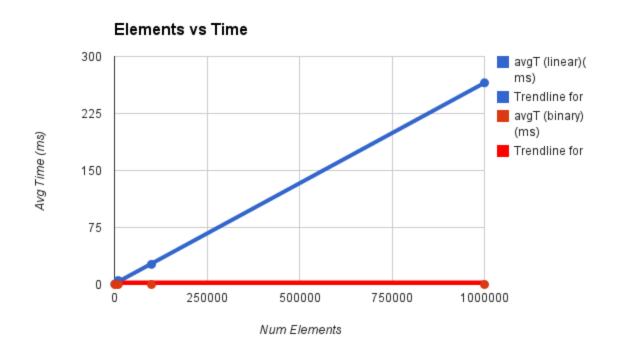
I ran tests to see if linear_search and binary_search ran according to their theoretical time.

linear_search, since it is looking through the list of elements one at a time, should theoretically have a running time of O(n).

binary_search, since it is cutting the elements it is looking at by half each time, should theoretically have a running time of O(lg n).

I took five trials of n values ranging from 1000 to 1000000 and s values of 100 and average the results together. This average can be seen on figure 1.

figure 1-



As you can see. The programs run at theoretical time.