Files

search.h

```
class Search {
        private:
            int sizeOfArray, elementsToBeSearched, element, size, lo, hi, mid, *list,
            *searchArray;
            string fileName, number;
            bool found;
        public:
            Search(string fileName)
            int numOfSearches()
            bool linear(int index)
            bool binary(int index)
```

main.cpp

```
int main()
        ofstream outfile
        string inputFile
        bool answer
```

Method Summary

bool binary (int)
Returns a boolean value as to whether or not the value was found
bool linear (int)
Returns a boolean value as to whether or not the value was found
Search(string)
Creates two arrays, the first is used to store all the numbers that need to be collected. The second array contains the values we will be searching for.
int numberOfSearches()
Returns the size of the second array
int main()
The main takes in the name of the file you have created then creates an output file that it will write to. The main also calls the search methods and then writes to the output file as it conducts the search. Lastly, the main captures the run times of the binary search and the linear search separately.

Method Detail

Search

Search(string name)

Opens the file specified by the user. Creates two arrays, the first is used to store all the numbers that need to be collected. The second array contains the values we will be searching for.

Throws:

Out of Memory exception

binary

public bool binary(int a)

Uses the already sorted array to quickly search and find if an element is in the array. Binary Search is able to eliminate half of the array each time it tries to search based on the numeric value of the element.

Parameters:

a - the index position you are at in the array that contains the list of elements you are searching for

Returns:

True or false depending on whether the element was found or not

linear

public bool linear(int a)

Uses the already sorted array but has to traverse every element in the array until the element is found. Worst case the element is not in the array and the whole array must be traversed before reaching this conclusion.

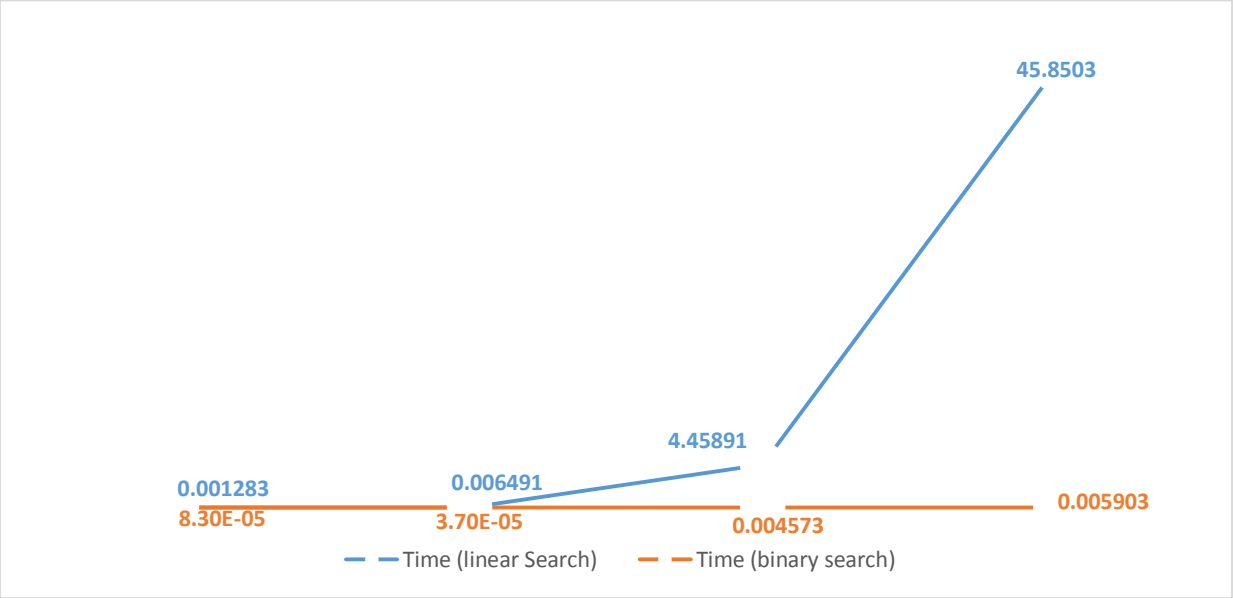
Parameters:

a - the index position you are at in the array that contains the list of elements you are searching for

Returns:

True or false depending on whether the element was found or not

Time Complexity Graph



n	Time (linear Search)	Time (binary search)
1,000	0.001283	8.30E-05
10,000	0.006491	3.70E-05
100,000	4.45891	0.004573
1,000,000	45.8503	0.005903