

DSA – Practical: Binary Search Evaluation

Refer to the notes on binary search and work on the following tasks:

1. Write the implementation of the two functions: sequentialSearch() and binarySearch():

```
//Sequentially searches the given array to find the index of value.
Returns -1 if value not found
int sequentialSearch (int arr[], int n, int value){
    ...
}

//Binary search algorithm to find index of given value in the sorted array.
Returns -1 if value not found
int binarySearch (int arr[], int n, int value){
    ...
}
```

2. Evaluate and compare the efficiencies of sequential and binary search algorithms. The array must be sorted first before using binary search. Because both search algorithms are very fast, use clock() function instead of time(NULL) to measure the time consumed by searching computations in milliseconds. clock() returns the number of clock ticks elapsed since the program was launched. Calling it before and after the computations, subtracting the obtained values, and then dividing the difference by (CLOCKS_PER_SEC/1000) will give the time consumed in milliseconds.

You might still notice that both algorithms require very little time (maybe less than 1 ms) to search for non-existing value (worst-case scenario). Therefore, instead of performing one search operation, we can perform the same operation in a loop for 100,000 times.

```
clocksBefore = clock();
for(i = 0; i<100000; i++)//search for a value that doesn't exist 100,000 times.
    sequentialSearch(arr1, MAX_SIZE, -99);
clocksAfter = clock();
timeInMS = (clocksAfter - clocksBefore)/(CLOCKS_PER_SEC/1000);
printf("Time for 100,000 sequential searches is %d milliseconds.\n", timeInMS);
```

Record experiment results in the given table below and discuss your findings:

Array size	Time (ms) consumed in 100,000 sequential searches (worst case for each)	Time (ms) consumed in 100,000 binary searches * (worst case for each)
1000		
10,000		
100,000		
...		

* Time needed to sort the array in order to apply binary search is not part of the comparison.

3. Fill up the same table again but after taking into account the time required to sort the array before using binary search (use any sorting algorithm you have implemented). Discuss your findings.

Array size	Time (ms) consumed in 100,000 sequential searches (worst case for each)	Time (ms) consumed to sort the array using selection sort and 100,000 binary searches (worst case for each)
1000		
10,000		
100,000		
...		

- The End -

