

Algorithm Design and Analysis

Thomas Kwashnak

Fall 2021

Contents

1 Mathematical Background

2 Searching

2.1 Sequential Search

In an unsorted array, one of the easiest sorting techniques is to just sequentially look through every index to find the item.

```
static int sequential(int[] A, int K) {  
    for (int i = 0; i < A.length; i++) {  
        if (A[i] == K) {  
            return i;  
        }  
    }  
    return A.length;  
}
```

While this also works on a sorted array, there is another much-easier method of searching through a sorted array:

2.2 Binary Search

If we know that an array is sorted, then we can perform what is called Binary Search. The idea of binary search is to ask a "higher or lower?" question with a pivot in the array, such that it reduces half of the array every time.

The pseudocode is as follows

Algorithm 1: Binary Search on a Sorted Array

Input: A sorted array: A , and the integer you wish to find: k

Output: The index i where k can be found in array A , or -1 if k is not in A

low = 0

high = $A.length - 1$

while $low \leq high$ **do**

 mid = $(low + high) / 2$

if $A[mid] < k$ **then**

 low = mid + 1

else if $A[mid] > k$ **then**

 high = mid - 1

else

return mid

end

return -1

3 Algorithm Analysis

4 Linear Structures

5 Recursion

6 Design

7 Binary Trees

8 Sorting

9 Hashing

10 General Trees

11 Graphs

12 Advanced Analysis Techniques

13 Search Structures

14 Advanced Algorithm Techniques