

DAA ASSIGNMENT 4

Group No 1

Pradhuman Singh Baid (IIB2019001)

Harsh Mahajan (IIB2019002)

TAUHID ALAM (BIM2015003)

Contents -

- ▶ ● Problem Statement
- ▶ ● Introduction
- ▶ ● Proposed Algorithm
- ▶ ● Time and space complexity analysis
- ▶ ● Conclusion
- ▶ ● References

Problem Statement

- Write a program to find if a given integer x appears more than $n/2$ times in a sorted array of n integers. Solve using divide and conquer algorithm.

Introduction

Divide and conquer is the way of solving a problem by dividing the problem into smaller sub-problems.

It is a technique in which the main problem is divided into smaller sub-problems and then solving the smaller sub-problems and combining them to find the solution of the problem.

In this paper, we have to find if a given integer x appears more than $n/2$ times in a sorted array of n integers

ALGORITHMIC DESIGN

In the problem we need to find whether a given integer x is appearing more than $n/2$ times in a sorted array or not. Efficiently this problem can be solved using binary search which is a divide and conquer algorithm.

First We divided the array in two parts and checked the middle element. If this element is greater than or equal the integer x then we would focus on the left sub-array otherwise right sub-array.

Now, the size of focused sub-array is reduced to half of original array .And then we will recursively apply this approach on the focused array until its size is reduced to 1.

When the size of resultant array is '1' it means that we are on the index where the integer x appeared 1st time. Let name that index as 'left'.

Now, we need to check whether it appeared more than $n/2$ times or not. For that we would add $n/2$ to 'left' and name the result 'right'. As it is provided that array is sorted and if 'x' appeared more than $n/2$ times then it must also present at index 'right'.

And finally we will print the result according to its presence on index 'right'

Time complexity

In this algorithm at each iteration the array is divided by 2. Suppose length of array is n and after K iteration the length of array becomes 1 and then we get

$$(n/2)^k = 1$$

$$n = 2^k$$

$$\log(n) = \log(2^k)$$

$$\log(n) = k \log(2)$$

$$k = \log(n)$$

Therefore, Time complexity is $O(\log(n))$

Space complexity

We need space for the input array of size n only, and for rest of the variables

we are using constant amount of space so

Space Complexity : $O(n)$

Conclusion

Through this assignment we conclude that fastest way to find an element is using divide and conquer algorithm.

References

- ▶ [1] Wikipedia: Divide-and-conquer algorithm,
https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm
- ▶ [2] GeeksforGeeks: Divide and Conquer Algorithm — Introduction,
<https://www.geeksforgeeks.org/divide-and-conquer-algorithm-i>