

COURSE NAME: ALGORITHM ANALYSIS

COURSE INSTRUCTOR: Dr. M. Amaç GÜVENSAN

STUDENT NAME: Ertuğrul ŞENTÜRK

STUDENT NO: 18011028

STUDENT E-MAIL: mdesenturk@gmail.com

SCHOOL YEAR: 3

GROUP NUMBER: 1

HOMEWORK NUMBER: 1

QUESTION NUMBER: 1

QUESTION: Find two closest number in an unordered array

ALGORITHM:

a-) In question a for a brute-force solution;

* Compared all the elements of the array in pairs.
* The absolute value of the difference between two elements is calculated.
* Kept the indices of the two elements with the smallest difference in absolute value, and then these elements printed as an output.

b-) For the optimal solution in question b;

* Array is sorted with merge sort algorithm.
* The elements of the listed array are compared with the next element in order.
* The elements with the lowest difference are outputted to the user.

ALGORITHM ANALYSIS:

a-) For brute force solution in case A;

Since there are 2 nested for loops, we choose the most repetitive part of the algorithm as the "if (diff <min)" comparison within the for loops. Since this comparison is done (n \* n) / 2 times in all cases, the complexities of worst, best and average case are equal. In this case, too, the complexity is θ(n2).

b-) For the optimal solution in case B;

In this algorithm, we navigate the merge sort algorithm and then the array with just a loop. Since the complexity of the sort algorithm will be higher and the merge sort algorithm is a divide and conquer algorithm, we can apply the master's theorem.

Since the most complex place in this algorithm is the merge part, if we take the comparison in the loop as the most frequently repeated event:

Cworst(n) = 2 \* Cworst(n / 2) + n-1; when n> 1 and C(1) = 0.

From here, according to the master theorem, a = 2, b = 2, d = 1. According to this result, since a = bd, the complexity is calculated as θ(n \* log (n)).

PROGRAM OUTPUTS:

1-) I used numbers from 10 to 1 as input.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldua)

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

b)

The main difference between the two algorithms is if there is more than one case in which the difference between the two arrays is the smallest, algorithm 1 gives the first two smallest elements according to the given order of the array. As the second algorithm ranks the array, it gives the numbers with the smallest difference and value.

2-) I used a 150-element array with numbers between 0 and 200,000 as input.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldua)

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu b)