Dashboard / My courses / CS23331-DAA-2023-CSE / Greedy Algorithms / 5-G-Product of Array elements-Minimum

Started on	Tuesday, 3 September 2024, 1:33 PM
State	Finished
Completed on	Tuesday, 3 September 2024, 2:53 PM
Time taken	1 hour 19 mins
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Given two arrays array_One[] and array_Two[] of same size N. We need to first rearrange the arrays such that the sum of the product of pairs(1 element from each) is minimum. That is SUM (A[i] * B[i]) for all i is minimum.

For example:

Input	Result
3 1 2 3 4 5 6	28

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>
int compareAsc(const void *a, const void *b) {
    return (*(int *)a - *(int *)b);
}
int compareDesc(const void *a, const void *b) {
    return (*(int *)b - *(int *)a);
}
int minProductSum(int array_One[], int array_Two[], int size) {
    qsort(array_One, size, sizeof(int), compareAsc);
    qsort(array_Two, size, sizeof(int), compareDesc);
    int minSum = 0;
    for (int i = 0; i < size; i++) {
        minSum += array_One[i] * array_Two[i];
    }

    return minSum;
}

int main() {
    int size;
    scanf("%d", &size);
    int *array_One = (int *)malloc(size * sizeof(int));
    int *array_Two = (int *)malloc(size * sizeof(int));

    if (array_One == NULL || array_Two == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }
    for (int i = 0; i < size; i++) {
        scanf("%d", &array_One[i]);
    }
    for (int i = 0; i < size; i++) {
        scanf("%d", &array_Two[i]);
    }
    int result = minProductSum(array_One, array_Two, size);
    printf("%d" ,result);
    free(array_One);
    free(array_Two);

    return 0;
}

```

	Input	Expected	Got	
✓	3 1 2 3 4 5 6	28	28	✓
✓	4 7 5 1 2 1 3 4 1	22	22	✓
✓	5 20 10 30 10 40 8 9 4 3 10	590	590	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◄ 4-G-Array Sum max problem

Jump to...

1-Number of Zeros in a Given Array ►