CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Quiz navigation

1

Finish review

Started on	Tuesday, 27 August 2024, 2:01 PM
State	Finished
Completed on	Tuesday, 27 August 2024, 2:49 PM
Time taken	47 mins 49 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Flag question

A person needs to eat burgers. Each burger contains a count of calorie. After eating the burger, the person needs to ru
If he has eaten i burgers with c calories each, then he has to run at least $3^i * c$ kilometers to burn out the calories.
burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are $(3^0 * 1) + (3^1 * 3) + (3^2$
But this is not the minimum, so need to try out other orders of consumption and choose the minimum value. Determine the
he needs to run. Note: He can eat burger in any order and use an efficient sorting algorithm.Apply greedy approach to s

Input Format

First Line contains the number of burgers

Second line contains calories of each burger which is n space-separate integers

Output Format

Print: Minimum number of kilometers needed to run to burn out the calories

Sample Input

```
3
5 10 7
```

Sample Output

```
76
```

For example:

Test	Input	Result
Test Case 1	3 1 3 2	18

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<math.h>
int main(){
    int n,temp=0,result=0;
    scanf("%d",&n);
    int a[20];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++){
        for(int j=0;j<n-1;j++){
            if(a[j]<a[i]){
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }

    for(int i=0;i<n;i++){
        result=result+(pow(n,i)*a[i]);

    }
    printf("%d",result);

}
```

	Test	Input	Expected	Got	
✔	Test Case 1	3 1 3 2	18	18	✔
✔	Test Case 2	4 7 4 9 6	389	389	✔
✔	Test Case 3	3 5 10 7	76	76	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

⇥ 2-G-Cookies Problem

Jump to...

4-G-Array Sum max problem ⇤