Problems

Courses Problem Editorial Submissions Comments

Job Fair Live

Cake Distribution Problem

185

Hard Accuracy: **64.62%** Submissions: **19K+** Points: **8**

Contests

POTD

Geek is organizing a birthday party, so his friends brought a cake for him. The cake consists of **N** chunks, whose individual sweetness is represented by the **sweetness** array. Now at the time of distribution, Geek cuts the cake into **K + 1** pieces to distribute among his **K** friends. One piece he took for himself. Each piece consists of some consecutive chunks. He is very kind, so he left the piece with **minimum** sweetness for him.

You need to find the **maximum** sweetness that the Geek can get if he distributes the cake optimally.

Example 1:

Input:
N = 6, K = 2
sweetness[] = {6, 3, 2, 8, 7, 5}
Output:
9
Explanation:
Geek can divide the cake to [6, 3], [2, 8], [7, 5]
with sweetness level 9, 10 and 12 respectively.

Example 2:

Input:
N = 7, K = 3
sweetness[] = {1, 2, 4, 7, 3, 6, 9}
Output:
7
Explanation:
Geek can divide the cake to [1, 2, 4], [7], [3, 6], [9]
with sweetness level 7, 7, 9 and 9 respectively.

Menu

Your Task:

You need to complete the **maxSweetness()** function which takes an integer array of **sweetness**, an integer **N** and an integer **K** as the input parameters and returns an integer denoting the maximum sweetness that the Geek can get.

Expected Time Complexity: O(NlogM), where M is the sum of elements in the array.

Expected Space Complexity: O(1)

Constraints:

$1 <= N <= 10^5$

$0 <= K < N$

$1 <= sweetness[i] <= 10^9$

View Bookmarked Problems

Topic Tags

C++ (g++ 5.4) Start Timer

```
// } Driver Code Ends
// User function Template for C++

#define ll long long int
bool check(ll mid, vector<int> &arr, int N, int K)
{
    ll sum = 0;
    int count = 0;
    // Aim - Number of pieces > = k
    for (int i = 0; i < N; i++)
    {
        sum += arr[i];
        if (sum >= mid)
        {
            count++;
            sum = 0;
        }
    }
    return count >= K + 1;
}

class Solution{
    public:

    int maxSweetness(vector<int>& sweetness, int N, int K) {
    // Write your code here.
    ll start = 1, end = (ll)(1e14);
    ll ans = -1;
    while (start <= end)
    {
        ll mid = (start + end) / 2;
```

Menu

Custom Input Compile & Run Submit

Report An Issue

If you are facing any issue on this page. Please let us know.

Menu