Jesse and Cookies

locked

by vatsalchanana

Problem Submissions Leaderboard Discussions Editorial

Editorial by vatsalchanana

This problem can be solved using a min heap. Initially, we add all the cookies to the heap. We repeatedly pop 2 cookies with the least sweetness and combine them and add the resulting sweetness ($1\times$ *least sweet cookie* + $2\times$ *2nd least sweet cookie*) to the heap till the sweetness of minimum becomes $\geq K$.

Statistics

Difficulty:	Easy
Time Complexity:	O(n * log(n))
Required Knowledge:	Priority Queue
Publish Date:	Jan 19 2016

Set by vatsalchanana

Problem Setter's code :

```
#include<iostream>
#include<vector>
#include<cstdio>
#include<algorithm>
#include<utility>
#include<set>
#include<map>
#include<cstring>
#include<cmath>
#include<string>
#include<cstdlib>
#include<queue>

using namespace std;

int main()
{
    #define int long long
    int n,k;
    cin>>n>>k;
    priority_queue<int, std::vector<int>, std::greater<int> > pq;
    for(int i=0;i<n;i++)
    {
        int val;
        cin>>val;
        pq.push(val);
    }
    int count=0;
    bool ans=true;
    while(1)
    {
        if(pq.empty())
        {
            ans=false;
            break;
        }
        int a1=pq.top();
        pq.pop();
        if(a1>=k)
        {
            break;
        }
        if(pq.empty())
        {
            if(a1<k)
            {
                ans=false;
            }
            break;
        }

        int a2=pq.top();
        pq.pop();

        int nv=a1+2*a2;
        count++;
        pq.push(nv);

    }
    if(ans)
        cout<<count;
    else
        cout<<"-1";
    cout<<endl;
}
```

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature