



分苹果



西南大学附属中学
High School Affiliated to Southwest University

递推里面的第二类斯特林数问题

模型：m个球，n个盘子，盘子允许为空

递归式： $f(m,n)=f(m,n-1)+f(m-n,n)$;

边界条件自行根据递归式完善

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



独木舟核心代码



西南大学附属中学
High School Affiliated to Southwest University

```
int a[500];
int cmp(int x,int y){
    return x>y;
}
int main(){
    int w,n,t;
    cin>>t;
    while(t--){
        int cnt=0;
        cin>>w>>n;
        for(int i=0;i<n;i++){
            cin>>a[i];
            sort(a,a+n,cmp); //从大到小排序

            int i=0,j=n-1;
            while(i<=j){
                if(a[i]+a[j]<=w){i++;j--;cnt++;}
                else if(a[i]==w){cnt++;i++;}
                else {i++;cnt++;}
            }
            cout<<cnt<<endl;
        }
        return 0;
    }
}
```



西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



最大数代码



西南大学附属中学
High School Affiliated to Southwest University

```
int cmp(string x,string y){  
    if(x+y>y+x)    return 1;  
    if(x+y<y+x)    return 0;  
}  
int main(){  
    int n,i;  
    cin>>n;  
    for(i=1;i<=n;i++) cin>>a[i];  
    sort(a+1,a+n+1,cmp);  
    for(i=1;i<=n;i++) cout<<a[i];  
    return 0;  
}
```

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



贪心策略：找到最接近的导弹系统去拦截

正确性：显然，如果用最接近的系统去拦截，才能有更大拦截其他高度的可能

```
int a[1010]; // 导弹高度
int l[1010]; // l[i]: 第i套拦截系统的最低高度
int main(){
    int n=1;
    while(cin>>a[n]) n++;
    int k=1; // 拦截导弹系统数
    int p,j=1;
    l[k]=a[1]; // 第一套导弹的高度为第一颗导弹的高度
    for(int i=2;i<=n;i++){
        p=0; // 每次循环p=0
        for(int j=1;j<=k;j++){ // 循环匹配系统，找到最符合的系统
            if(l[j]>=a[i]){
                if(p==0) p=j; // 表示找到了其中一套符合要求的
                else if(l[p]>l[j]) p=j; // 如果同时满足例如l[1]=155>65, l[2]=158>65, 则应该应将最小值赋值给最小拦截系统
            }
        }
        if(p==0){
            k++; // 如果发现a[i]>l[j], 则应该增加一个系统
            l[k]=a[i]; // 更新新系统的最小值
        }
        else l[p]=a[i]; // 更新原系统的拦截最小值
    }
    cout<<k<<endl;
    return 0;
}
```



均分纸牌



西南大学附属中学
High School Affiliated to Southwest University

题意：给出一堆纸牌，现在你可以每次取若干张纸牌，最后需要你让每一堆都平均，问最少的移动次数

这道题难点在于贪心策略的发现

一堆牌： 3 7 17 13
需求： -7 -3 7 3

关键点在于不要有多余的移动，如何实现？

在这里，我们可以把负数的牌也看成实实在在的牌进行移动



均分纸牌



西南大学附属中学
High School Affiliated to Southwest University

```
#include<bits/stdc++.h>
using namespace std;
const int Maxn = 105;
int main(){
    int N, i, a[Maxn], b = 0, cnt = 0;
    cin >> N;
    for(i=0; i<N; i++){
        cin >> a[i];
        b += a[i];
    }
    b /= N; //计算平均数
    for(i=0; i<N; i++) a[i]-=b; //每堆牌都减去牌数
    for(i=0; i<N; i++){
        if(a[i] == 0)continue; //已经满足均分的牌，不需要再移动
        cnt++;
        a[i+1] += a[i];
    }
    cout << cnt;
    return 0;
}
```

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



大神排队



西南大学附属中学
High School Affiliated to Southwest University

题目：现在共有 n 个同学要排成一列，每个同学有两个属性：影响力 a_i 和承受能力 b_i 。给一个同学造成的心理创伤等于所有在他前面的影响力之和减去他的承受能力。请安排一下排队顺序，尽量使受到心理创伤最大的同学少受创伤。

样例输入1

3

10 3

2 5

3 3

样例输出1

2



| 西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



sum

a1

a2

对于相邻两个人来说

前一个人的影响力为 a_1 , 承受能力为 b_1

后一个人的为 a_2, b_2 , 那么前面所有人的影响力总和为sum

位置交换前创伤分别为 $\text{sum} - b_1$ 和 $\text{sum} + a_1 - b_2$

位置交换后创伤分别为 $\text{sum} - b_2$ 和 $\text{sum} + a_2 - b_1$

都减去一个sum

所以应该比较 $\max(-b_1, a_1 - b_2)$ 和 $\max(-b_2, a_2 - b_1)$

显然 $a_2 - b_1 > -b_1, a_1 - b_2 > -b_2$,

所以如果 $a_1 - b_2 > a_2 - b_1$, 交换后更优, 反之不交换更优

确定贪心策略: 结合影响力和承受力, 从交换前后的值决定如何排序

Thanks

For Your Watching

