codeforces round 643(div2)

A

题意：给定一个数字x□将它按照十进制分成许多位，组成一个集合，每次都挑选集合中最大值与最小值，将两者乘积加在x上，问k此操作后为多少？

题解：显然经过有限次操作之后，可以使最小位变成0，之后无论怎么操作都是0。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
ll work(ll n)
{
    int mx=0,mn=11;
    ll cun=n;
    while(n)
    {
        int now=n%10;
        mx=max(mx,now);
        mn=min(mn,now);
        n/=10;
    }
    return cun+mx*mn;
}
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        ll n,m;
        scanf("%lld%lld",&n,&m);
        m-=1;
        while(m--)
        {
            ll st=n;
            n=work(n);
            if(n==st)
            {break;
            }
        }
        printf("%lld\n",n);
    }
 }
```

B

题意：给一段数字，将这些数字分组，在分配完之后，规定数i所在的集合必须满足个数≥i,问最多能分出多少个组。

题解：经过枚举发现，有一种方案一定是最优的，将所给数排序，将值为i的数以i个为集合容量放在一起，如果无法完整的分成整数的个数集合，则把剩余的个数加在下一个数上，以此类推。

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=2e5+13;
int t,n;
int a[maxn];
int cnt[maxn];
int main()
{
    scanf("%d",&t);
    while(t--)
    {
        int ans=0;
        scanf("%d",&n);
        for(int i=1;i<=n;i++) cnt[i]=0;
        for(int i=1;i<=n;i++) scanf("%d",&a[i]),cnt[a[i]]++;
        sort(a+1,a+1+n);
        int res=0;
        for(int i=1;i<=n;i++)
        {
            if(cnt[i])
            {
                cnt[i]+=res;
                ans+=cnt[i]/i;
                res=cnt[i]%i;
            }
        }
        printf("%d\n",ans);
    }
}
```

≤

C

题意：给定四个数A□B□C□D□问有多少个三角形满足三条边a,b,c□满足A≤a≤B≤b≤C≤c≤D□

题解：众所周知三角形成立条件为a+b>c,那么只要满足a+b>c即可满足成立三角形，然后我们枚举c的每一个可能的值，列出限制条件，利用线性规划即可求出取值范围，最后要注意判断取并后是否合法。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
ll ans=0;
int main()
{
    ll ans=0;
    int a,b,c,d;
    scanf("%d%d%d%d",&a,&b,&c,&d);
    for(int i=c+1;i<=b+c;i++)
    {
        int l=max(b,i-b);
        int r=min(c,i-a);
        int now=min(i-1,d);
        if(r>=l) ans=ans+1ll*(r-l+1)*(now-c+1);
    }
    printf("%lld",ans);
}
```

D

题意:a和b在玩一个游戏，a需要选取n个数数使得这些数的和正好为w,并且a需要选取一个数k小于w，b需要从当中选取一个子序列使子序列的和为k，如果b能做到，则b赢，否则a赢，其中w和n为给定的。

题解：发现，如果w≥2n,则a必定可以使得选出来的序列中最小值大于等于2，则这时选取k为2时，则一定可以让a赢。否则一定找不到k使得a必赢，证明~~不是显然吗……~~

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=1e6+13;
int a[maxn];
int main()
{
    int n,s;
    scanf("%d%d",&n,&s);
    if(s/n>=2)
    {
        for(int i=1;i<=n;i++) a[i]=0;
        printf("Yes\n");
        int now=s/n;
        for(int i=1;i<=n;i++)
        {
            if(i==n)
            {
                a[i]=s;
                break;
            }
```

```
            a[i]=now;
            s-=now;
        }
        for(int i=1;i<=n;i++) printf("%d ",a[i]);
        puts("");
        printf("1");
     }
     else printf("No");
}
```

E

题意：给定一个序列，有三种操作，第一种以a为代价让序列中一个数-1，第二种操作为以b为代价让序列种一个数+1，第三种操作是以c为代价让一个数+1，同时让一个数-1。问要将序列中所有数都变成一样的最少需要花费多少代价。

题解：首先和容易想到枚举最终的数，肯定不能暴力枚举，所以要考虑答案随最终的数变化的轨迹，发现如果最终高度很小，则花费很大，最终高度很大，花费同样很大，最小值一定在中间某点取得，而且变化近似为二次曲线关系，则容易想到三分法，现在考虑给定高度，计算花费，首先可以意识到一点，一次第三种操作可以抵消一次第一，二种操作，接下来只要比较a+b和c哪个大，即可解决问题。

```
#include <bits/stdc++.h>//凡是函数变化关系呈现峰形则可以用三分法求解最值
using namespace std;
typedef long long ll;
const int maxn=1e5+13;
ll t[maxn];
int a,r,m,n;
ll check(ll now)
{
    ll ans=0;
    ll up=0,down=0;
    for(int i=1;i<=n;i++)
    {
        if(t[i]<now)
        {
            up+=now-t[i];
        }
        else{
            down+=t[i]-now;
        }
    }
    ans=1ll*r*down+1ll*up*a;
    ll res=min(up,down);
    if(a+r>m)
    {
        ans+=1ll*res*m-1ll*(a+r)*res;
    }
    return ans;
```

```
}
int main()
{
    ll ans=0;
    scanf("%d%d%d%d",&n,&a,&r,&m);
    for(int i=1;i<=n;i++)
    {
        scanf("%lld",&t[i]);
    }
    int l=0,r=1e9+13;
    while(l<r)
    {
        int ml=l+(r-l)/3,mr=r-(r-l)/3;
        ll e=check((ll)ml),f=check((ll)mr);
        if(e<f)
        {
            r=mr-1;//在区间的左边
        }
        else
        {
            l=ml+1;//在区间的右边
        }
    }
    ans=check((ll)l);
    printf("%lld",ans);
 }
```