

codeforces educational round 87(div2)

A

题意：大模拟

题解：题目比较长，理清逻辑关系即可，没什么好说的……

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int t,a,b,c,d;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d%d%d%d",&a,&b,&c,&d);
        if(b>=a)
        {
            printf("%d\n",b);
            continue;
        }
        else
        {
            int res=a-b;
            if(d>=c)
            {
                printf("-1\n");
                continue;
            }
            else
            {
                int t=ceil((double)res/(c-d));
                printf("%lld\n",1ll*t*c+1ll*b);
            }
        }
    }
}
```

B

题意：给定一个只由123组成的字符串，问取该字符串的一段连续子串，至少要取多长的子串才能满足子串中既有1也有2还有3。

题解：首先判断是否三个数都出现了，若没有都出现，直接就可以判定不成立。然后，最不动脑经的做法，就是从第一个开始模拟，找出每一段连续的且同为一个字符的串，观察该串开头的左边和末尾的右边的字

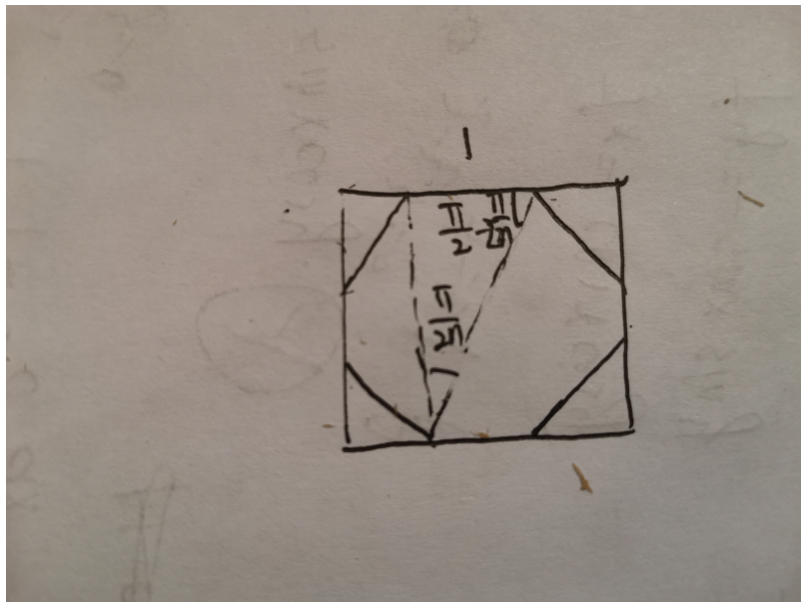
符是否为同一个，若不是则为一种情况。复杂度 $O(n)$

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=2e5+13;
char s[maxn];
int flag[4];
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        for(int i=1;i<=3;i++) flag[i]=0;
        int ans=0x7fffffff;
        int yes=1;
        scanf("%s",s+1);
        int len=strlen(s+1);
        for(int i=1;s[i];i++)
        {
            flag[s[i]-'0']=1;
        }
        for(int i=1;i<=3;i++)
        {
            if(!flag[i]) yes=0;
        }
        if(!yes) {
            printf("0\n");continue;}
        int k=1,st;
        while(s[k]==s[1]) k++;
        st=k-1;
        for(int i=k;i<=len;i++)
        {
            int j=i;
            while(s[j]==s[i]) j++;
            if(j>len) {
                break;}
            if(s[j]!=s[st]) ans=min(ans,j-i+2);
            st=j-1;
            i=st;
        }
        printf("%d\n",ans);
    }
}
```

C1

题意：给一个边长全部都为1的正 $2n$ 边形，这里 n 取偶数，要求一个正方形完全盖住这个正 $2n$ 边形，问最小边长是什么。

题解：如果 n 是偶数我们能够比较容易的看出合法的情况为下图



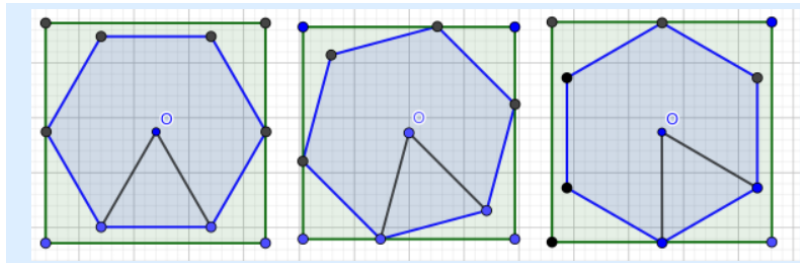
则很容易得出答案为 $\frac{1}{\tan \frac{\pi}{2n}}$

```
#include <bits/stdc++.h>
using namespace std;
const double pi=3.1415926535;
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int n;
        scanf("%d",&n);
        n*=2;
        double ans=1.0/(tan(pi/((double)n)));
        printf("%.10lf\n",ans);
    }
}
```

C2

题意：将上述题的 n 改成奇数

题解：如图



利用对称性，发现当 n 是奇数时我们可以发现当正多边形在旋转时具有一个对称性，我们旋转 $\frac{\pi}{2n}$ 时，图形会变为原来的状态。则我们不难发现当旋转为一半，即为 $\frac{\pi}{4n}$ 可以达到最大，那么我们有几何关系可以得到答案为 $\frac{\cos\frac{\pi}{4n}}{\sin\frac{\pi}{2n}}$

```
#include <bits/stdc++.h>
using namespace std;
const double pi=3.1415926535;
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int n;
        scanf("%d",&n);
        double x=pi/(4.0*n);
        double ans=1.0/(2*sin(x));
        printf("%.10lf\n",ans);
    }
}
```

D

题意：给一系列数，每个数都满足小于等于 n 每次可能进行两个操作，一种操作为将 k 插入序列，第二种操作为将第 k 个数消去。问 k 个操作后是否还有数。

题解：注意一下空间限制，这个限制不能用平衡树，不然死的很惨……，可以利用权值线段树，空间复杂度 $O(2n)$ 可以满足。

```
#include <bits/stdc++.h>
#define mid (l+r)/2
const int maxn=1e6+13;
using namespace std;
int p=0;
int lc[maxn<<1],rc[maxn<<1];
int val[maxn<<1];
void modify(int l,int r,int &rt,int lo,int add)
{
    if(!rt) rt=++p;
    if(l==r) val[rt]=add;
    else
    {
        if(lo>mid) modify(mid+1,r,rc,lo,add);
        else modify(l,mid,lc,lo,add);
    }
}
```

```
    if(l==r)
    {
        val[rt]+=add;
        return;
    }
    if(lo<=mid) modify(l,mid,lc[rt],lo,add);
    else modify(mid+1,r,rc[rt],lo,add);
    val[rt]=val[lc[rt]]+val[rc[rt]];
}
int find(int l,int r,int rt,int k)
{
    if(l>=r) return l;
    if(val[lc[rt]]>=k)
    {
        return find(l,mid,lc[rt],k);
    }
    else
    {
        return find(mid+1,r,rc[rt],k-val[lc[rt]]);
    }
}
int main()
{
    int rt=0,n,m,cnt=0;
    int x;
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;i++)
    {
        scanf("%d",&x);
        modify(1,n,rt,x,1);
        cnt++;
    }
    for(int i=1;i<=m;i++)
    {
        scanf("%d",&x);
        if(x>0)
        {
            modify(1,n,rt,x,1);
            cnt++;
        }
        else
        {
            int k=find(1,n,rt,-x);
            modify(1,n,rt,k,-1);
            cnt--;
        }
    }
    if(cnt==0) printf("0");
    else printf("%d",find(1,n,rt,1));
}
```

From:
<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:codeforces_educational_round_87_div2 

Last update: **2020/05/24 21:21**