

问题 A: Just another board game

/\*

贪心:

如果  $k=1$ , 则取第一行的最大值, 否则分奇偶讨论。

如果  $k$  为奇数, 则最后一次是一行中取最大值, 答案为每行最大值的最小值。

如果  $k$  为偶数, 则最后一次是一列中取最小值, 答案为每列最小值的最大值。

特殊情况, 如果  $g[1][1]$  的值更大, A 可以选择一开始就游戏结束。

\*/

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int n,m,g[100005];
```

```
long long k;
```

```
void solve(){
```

```
    scanf("%d%d%lld",&n,&m,&k);
```

```
    for(int i=1; i<=n; i++)
```

```
        for(int j=1; j<=m; j++)
```

```
            scanf("%d",&g[(i-1)*m+j]);
```

```
    int ans=0;
```

```
    if(k==1){
```

```
        for(int i=1; i<=m; i++) ans=max(ans,g[i]);
```

```
        printf("%d\n",ans); return;
```

```
    }
```

```
    if(k%2==1){
```

```
        ans=1e9;
```

```
        for(int i=1; i<=n; i++){
```

```
            int tmax=0;
```

```
            for(int j=1; j<=m; j++)
```

```
                tmax=max(tmax,g[(i-1)*m+j]);
```

```
            ans=min(ans,tmax); //每行最大值最小
```

```
        }
```

```
    }
```

```
    else{
```

```
        for(int j=1; j<=m; j++){
```

```
            int tmin=1e9;
```

```
            for(int i=1; i<=n; i++)
```

```
                tmin=min(tmin,g[(i-1)*m+j]);
```

```
            ans=max(ans,tmin); //每列最小值最大
```

```
        }
```

```
    }
```

```
    printf("%d\n",max(ans,g[1])); //可能 g[1][1]更大
```

```
}
```

```
int main(){
```

```
    int T; scanf("%d",&T);
```

```
    while(T--) solve();
```

```

        return 0;
    }

```

问题 B: 给树加边

/\*

如果没有额外的一条边，每个点能访问的点数，就是每个点  $i$  子树点数数量  $sz[i]$

选择肯定会选择某个叶子连到根，对于树上的点  $i$ ，如果  $i$  点所在的链和根形成环，原先  $i$  点访问数量  $sz[i]$  变为  $n$ ，增加了  $n-sz[i]$

选择增加值最大的链即为答案。

\*/

```

#include<bits/stdc++.h>

```

```

using namespace std;

```

```

typedef long long ll;

```

```

const int M=500005;

```

```

int n,tot,head[M],Next[M*2],vet[M*2],sz[M];

```

```

ll ans,maxs;

```

```

void add(int a,int b){

```

```

    Next[++tot]=head[a], vet[tot]=b;

```

```

    head[a]=tot;

```

```

}

```

```

void dfs(int x){

```

```

    sz[x]=1;

```

```

    for(int i=head[x]; i;i=Next[i]){

```

```

        int y=vet[i]; dfs(y);

```

```

        sz[x]+=sz[y];

```

```

    }

```

```

    ans+=sz[x];

```

```

}

```

```

void dfs2(int x,ll s){

```

```

    maxs=max(maxs, s);

```

```

    for(int i=head[x]; i;i=Next[i]){

```

```

        int y=vet[i];

```

```

        dfs2(y,s+n-sz[y]); //如果选择  $y$  分支连通根， $y$  点可以访问的点增加  $n-sz[y]$ 

```

```

    }

```

```

}

```

```

void solve(){

```

```

    scanf("%d",&n);

```

```

    tot=0; ans=0; maxs=0;

```

```

    for(int i=1; i<=n; i++){

```

```

        head[i]=0; sz[i]=0;

```

```

        if(i>1){

```

```

            int a; scanf("%d",&a);

```

```

            add(a,i);

```

```

        }

```

```

    }
    dfs(1);
    dfs2(1,0);
    printf("%lld\n",ans+maxs);
}
int main(){
    int T; cin>>T;
    while(T--) solve();
    return 0;
}

```

## 问题 C: 递增子序列和递减子序列

Hdu6852

### 题目大意

要你构造一个长为 $n$  ( $n \leq 1e5$ ) 的全排列，使得其最长上升子序列和最长下降子序列的长度分别为 $x$ 和 $y$ ，而且要使得这个构造的全排列，字典序最小，如果不能构造则输出NO，否则输出YES，并且输出其全排列

### 题目思路

这个构造比较神奇吧，以前都没有见到过，现在学习一下，其实他的构造方法简单来说就是**分段和翻转**

如本来是1 2 3 4 5 6 7 8 9 10的全排列 而你如果要构造一个长度为5的单调递减序列那么你就把最后5个元素翻转变成

1 2 3 4 5 10 9 8 7 6 然后假如你要构造4个最长的单调上升子序列，显然是前面5个数找三个出来，因为最后这5个数肯定可以有一个元素在最长上升子序列的最后。你就可以把前面5个数分为三段。然后再将这三段分别翻转，而且你的每一段的长度不能超过5，如果超过5了，那么最长下降子序列就会变长，而且你要使得这些分配中，前面段的长度尽可能小（为了保证字典序最小），那么这个排列就会分为1...2...3 4 5...6 7 8 9 10这四段，然后分别翻转变成1 2 5 4 3 10 9 8 7 6

我不太会证明这个字典序为何是最小的，但是感觉很像（ac就是证明了

然后如何判断是否成立呢，显然最后 $y$ 个元素翻转后还剩下 $n-y$ 个元素，要分为 $x-1$ 段，而且每一段都必须是大于一且小于等于 $y$ 即：

$$1 \leq \frac{n-y}{x-1} \leq y$$

$$x + y - 1 \leq n \leq x * y$$

然后自己贪心使得前面的段尽可能短，具体看代码实现

### 代码 1:

```
#include<cstdio>
```

```
int tes,n,a,b,len,num;
```

```

void solve(){
    scanf("%d%d%d",&n,&a,&b);
    if(a+b>n+1 || (long long)a*b<n){puts("NO");return;}
    puts("YES");
    if(b==1){for(int i=1;i<=n;i++)printf("%d ",i);puts("");return;}
    num=(n-a)/(b-1);
    len=(n-a)%(b-1)+1;
    if(len==1)len=0;
    for(int i=1;i<=n-num*b-len;i++)printf("%d ",i);
    for(int i=n-num*b;i>n-num*b-len;i--)printf("%d ",i);
    for(int k=n-(num-1)*b;k<=n;k+=b)
    for(int i=k;i>k-b;i--)printf("%d ",i);
    puts("");
}
int main()

```

```

{
    scanf("%d",&tes);
    while(tes--)solve();
    return 0;
}

```

代码 2:

```

#include<bits/stdc++.h>
using namespace std;
typedef long long LL;
typedef pair<int,int> PII;
const int N=100005;
int t,n,x,y;
int a,b,c,d,e;
bool pd;
int main()
{
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d%d%d",&n,&x,&y);
        if((LL)x*y<n | x+y>n+1)
        {
            puts("NO");
            continue;
        }
        puts("YES");
        if(x+y==n+1)
        {
            for(int i=1;i<x;i++) printf("%d ",i);
            for(int i=n,j=1;j<y;j++,i--) printf("%d ",i);
            printf("%d",x);
            puts("");
            continue;
        }
        pd=1;
        a=(n-x)/(y-1);
        b=n-x-a*(y-1)+1;
        c=n-a*y-b;
        for(int i=1;i<=c;i++) if(pd) printf("%d",i),pd=0;
        else printf(" %d",i);
        for(int i=c+b;i>max(0,c);i--) if(pd) printf("%d",i),pd=0;
        else printf(" %d",i);
        c+=b;
    }
}

```

```

        for(int j=1;j<=a;j++)
        {
            for(int i=c+y;i>c;i--)
            {
                if(pd) printf("%d",i),pd=0;
                else printf(" %d",i);
            }
            c+=y;
        }
        puts("");
    }
}

```

问题 D: 数字串

Hdu6831

## Fragrant Numbers

首先长度有一个明显的上界，即 $n$ ，因为每个位置数字都大于等于1，没有0的存在。

可以进行一个简单的dp,  $bool\ dp[L][R][val]$  表示  $[L, R]$  区间内是否能构造出 $val$ .

题目中括号、加号、乘号的添加，允许了 $dp[L][mid][val1]\ dp[mid+1][R][val2]$  向  $dp[L][R][val1+val2]$  以及  $dp[L][R][val1*val2]$  转移

由于长度上界为  $n$ ，我们需要枚举  $1 \sim n$  的所有  $L, R$ , 再加上两个值的枚举，复杂度  $O(n^4)$

简单验证小情况，发现，在 $n$ 较小的情况下，仅3和7无法构造出来。对于其他情况，答案都小于13

可以猜测长度的上界不大，我们不再将  $L$  和  $R$  枚举到  $n$ , 打表验证发现，如果假定长度上界为13,  $n \leq 5000$  范围内的所有 $n$ ，仅 3 和 7 无法出解，而这两个数确实是无法被构造出来的。我们可以确信，对于其他能在这个范围内出解的数，输出出来的解一定是长度最短的答案。

在我们缩小了  $L$  和  $R$  的范围后，复杂度缩小至 $O(13^2n^2)$ ，由于  $n$  较小，可以选择打表或者直接提交这个  $O(13^2n^2)$  的程序。(后者可能需要一定的常数优化)

代码 1:

```

#include <cstdio>
#include <iostream>
#include <algorithm>
#include <cstring>
#include <vector>
#include <queue>
#include <map>
#include <set>
#include <climits>
#include <cmath>
using namespace std;

```

```
int T,n;
```

```

set<int> dp[20][20];
string s="11451419191145141919";
int ans[5010];

void dfs(int l,int r)
{
    if (!dp[l][r].empty()) return;
    if (r-l+1<=4)
    {
        int t=stoi(s.substr(l-1,r-l+1));
        if (t<=5000) dp[l][r].insert(t);
    }
    for (int i=l;i<r;i++)
    {
        dfs(l,i);
        dfs(i+1,r);
        for (int x:dp[l][i])
        {
            for (int y:dp[i+1][r])
            {
                if (x+y<=5000) dp[l][r].insert(x+y);
                if (x*y<=5000) dp[l][r].insert(x*y);
            }
        }
    }
}

int main()
{
    //freopen("in.txt","r",stdin);
    //freopen("b.txt","w",stdout);
    ios::sync_with_stdio(0); cin.tie(0);
    memset(ans,-1,sizeof(ans));
    for (int i=1;i<=11;i++)
    {
        dfs(1,i);
        for (int x:dp[1][i])
        {
            if (ans[x]==-1) ans[x]=i;
        }
    }
    cin>>T;
    while (T--)
    {

```

```

        cin>>n;
        cout<<ans[n]<<"\n";
    }
    getchar(); getchar();
    return 0;
}

```

代码 2:

```
#include<cstdio>
```

```
int s[17]={0,1,1,4,5,1,4,1,9,1,9,1,1,4,5,1,4};
```

```
int
```

```

ans[5005]={0,1,2,-1,3,3,3,-1,3,4,4,2,5,4,3,3,6,6,4,4,4,4,4,5,4,4,5,6,6,4,4,5,6,6,5,5,6,6,6,4,5,5,7,3,
4,4,4,5,4,5,6,6,6,6,5,4,5,6,5,6,6,7,6,6,5,5,7,6,6,4,4,5,7,6,4,5,6,6,6,6,6,7,5,5,6,6,6,6,4,5,5,7,6,5,6,
6,6,4,5,6,6,6,6,6,6,6,7,5,6,7,6,3,7,7,7,6,4,5,6,7,6,6,6,6,6,7,6,7,7,6,6,6,7,6,6,6,6,7,6,6,4,4,5,6,
6,6,6,6,7,6,6,7,8,7,6,6,7,7,7,6,5,7,7,6,6,7,7,7,8,7,7,8,8,8,8,6,6,6,7,6,6,6,7,6,6,7,7,6,6,7,8,7,6,
7,7,8,5,5,5,8,6,6,6,7,6,7,7,5,7,8,6,6,4,5,6,7,6,6,6,6,6,7,6,5,7,8,6,6,6,7,8,8,7,7,8,8,8,6,7,7,6,7,7,6,
6,7,5,5,8,8,6,6,6,7,7,5,7,6,6,6,7,7,7,8,8,6,6,7,8,8,6,6,6,7,6,7,7,6,7,8,6,6,7,7,8,7,6,7,7,8,6,
7,5,8,6,7,6,7,8,8,8,7,7,8,6,6,7,8,7,8,7,7,8,8,8,6,7,8,6,7,8,8,8,8,7,8,8,6,6,7,9,8,6,7,
8,8,8,6,7,7,8,6,7,8,8,6,7,7,7,8,6,6,7,8,7,7,6,7,8,8,8,7,8,8,8,6,7,8,8,6,7,8,8,8,8,7,5,
8,7,7,6,7,8,8,6,7,7,8,6,7,8,8,8,7,7,8,7,8,8,8,8,6,7,8,8,8,8,6,7,7,8,8,8,9,7,7,6,5,5,8,6,6,6,7,8,7,
7,5,8,8,7,6,7,8,8,8,8,8,8,7,6,7,8,8,6,7,8,8,7,8,8,8,8,8,7,8,7,7,4,5,8,8,6,6,7,8,7,8,8,5,8,8,6,6,7,8,
8,8,8,8,8,6,6,6,7,6,7,8,8,8,8,6,6,7,8,8,7,8,8,7,7,8,8,8,9,8,7,7,8,8,6,7,7,8,8,
8,8,8,7,8,4,5,8,8,6,6,7,8,8,8,6,6,7,8,6,6,7,8,6,7,8,8,8,8,7,6,7,8,8,8,7,8,8,8,5,8,8,8,6,7,7,9,8,8,8,
8,8,8,8,8,8,8,8,9,8,6,7,6,6,6,7,8,8,7,8,8,8,8,6,7,8,6,6,7,8,8,6,7,8,9,8,8,8,8,8,6,7,6,7,8,8,8,8,8,8,
8,8,7,8,9,8,7,7,7,8,8,8,8,7,8,5,8,6,7,6,7,7,7,9,8,8,8,8,8,8,8,8,8,8,7,8,8,8,7,7,7,8,7,5,5,8,8,6,6,7,
8,8,8,8,6,6,7,8,8,6,7,9,9,8,7,7,8,7,8,8,8,8,9,8,8,9,8,8,7,8,8,9,8,8,7,7,8,8,8,8,8,9,8,5,8,8,8,6,6,7,
9,8,8,8,8,8,7,7,8,9,8,6,7,8,9,8,8,8,8,8,8,8,9,8,9,7,7,8,8,8,8,9,8,7,8,8,9,8,8,9,9,6,7,6,6,6,7,6,6,7,8,6,
6,7,6,7,8,8,7,8,8,8,8,6,7,8,8,7,7,7,8,8,8,8,7,8,8,8,7,8,8,8,8,8,9,6,7,7,7,8,8,8,8,9,7,7,7,8,8,8,7,
8,8,8,6,7,7,7,6,7,8,8,9,8,8,7,8,8,8,9,8,8,9,8,8,8,5,7,9,9,6,7,9,8,6,7,7,8,8,8,8,9,8,8,8,8,8,6,7,8,8,8,
8,8,8,8,8,8,7,8,8,8,8,9,8,7,8,8,7,8,8,9,8,8,8,9,8,8,8,7,8,9,8,8,8,8,9,9,8,8,8,8,8,8,9,8,8,9,9,8,8,6,
6,7,9,7,7,7,9,8,8,8,8,8,6,7,9,9,8,8,8,8,8,8,8,9,9,8,8,8,8,8,8,9,9,6,6,7,8,6,7,6,7,9,8,7,7,9,
8,8,8,6,7,7,8,8,8,9,8,8,8,8,8,8,6,7,8,8,8,8,6,7,8,8,8,8,8,9,8,7,7,7,9,8,8,8,8,9,8,7,8,9,9,9,8,8,8,8,
8,8,9,9,8,8,8,7,8,9,8,8,9,8,9,6,7,8,8,8,7,8,7,8,8,8,7,9,9,8,8,8,9,9,8,8,8,8,9,7,8,8,8,9,8,8,8,8,8,
9,10,9,6,7,9,8,7,4,5,8,8,6,6,7,8,8,8,7,9,9,8,6,7,8,9,9,8,9,8,9,7,8,8,8,9,9,9,8,8,8,8,8,8,8,8,7,7,
8,9,8,8,9,10,9,8,8,8,9,8,8,8,8,9,9,8,9,9,10,8,8,9,10,9,8,8,8,8,9,8,8,9,6,7,8,9,9,9,7,7,8,8,8,8,9,1
0,9,8,8,9,8,8,9,9,8,8,9,10,8,8,9,10,7,9,9,10,10,9,6,7,7,8,9,8,8,8,8,7,7,7,9,10,8,7,7,9,8,8,7,9,8,9,8,8
,7,8,7,7,8,8,8,9,8,8,8,8,8,8,8,8,7,10,9,8,9,9,9,8,9,8,7,9,8,8,8,9,8,8,7,8,8,9,10,8,8,9,8,8,8,8,9,8,8,
8,9,8,8,8,8,9,8,8,9,10,9,9,7,8,8,9,10,8,8,8,8,8,8,8,9,9,8,8,9,9,8,9,10,9,8,9,9,9,7,9,10,9,9,
9,6,7,8,8,8,9,10,9,10,8,8,7,9,10,8,8,9,9,8,8,8,8,8,9,7,7,9,10,8,8,9,8,7,8,8,8,9,9,8,8,8,8,8,8,8,9,
9,7,8,8,8,8,8,9,10,8,8,8,9,9,8,8,5,5,9,10,6,6,7,8,8,8,9,10,8,8,8,8,8,8,9,9,9,10,8,8,7,9,9,9,8,9,8,
9,9,7,8,8,8,8,8,7,7,8,8,8,8,8,9,8,8,8,8,9,9,9,8,8,8,8,8,8,8,9,9,8,8,8,8,8,9,9,8,8,9,7,8,8,8,8,
9,9,9,8,8,8,8,8,8,9,9,9,8,8,9,8,7,9,10,9,9,9,10,8,9,8,9,8,9,8,8,9,9,9,9,10,9,9,8,9,9,10,10,9,8,9,9
,8,9,10,10,10,10,8,8,9,10,10,8,8,8,8,9,8,9,9,10,10,9,8,8,9,10,8,8,9,10,10,8,8,8,8,9,8,8,8,9,9,9,8,9,
9,8,8,8,6,7,7,9,10,10,8,8,7,8,8,8,9,10,9,8,9,8,7,8,8,8,9,10,8,8,8,8,9,9,9,10,8,8,6,7,9,10,8,9,8,8,8,8,
8,9,9,10,7,7,8,8,8,8,9,9,9,8,8,8,8,8,8,9,8,8,8,8,8,8,8,8,8,8,9,8,9,10,8,8,8,9,9,9,8,9,10,8,9,9,7,9,9,8,

```

9,9,8,8,9,9,9,9,9,8,9,8,9,9,9,10,8,9,8,8,9,8,9,9,9,10,8,9,9,9,9,10,9,9,9,7,9,8,9,9,  
9,10,8,8,8,9,9,8,7,8,9,10,10,10,8,9,9,8,9,10,10,10,8,8,8,9,6,6,6,7,6,6,7,8,6,7,8,6,7,8,8,8,8,8,  
8,8,8,8,8,9,9,8,9,7,8,8,9,9,10,8,8,8,8,8,7,7,7,6,7,8,8,9,8,8,8,7,8,8,8,9,8,9,8,9,9,8,8,8,8,  
8,8,8,9,10,10,8,8,8,8,9,8,7,7,7,8,7,7,9,8,8,8,7,8,8,8,7,8,9,8,8,8,9,8,9,8,9,10,8,8,8,9,8,8,9,1  
0,9,8,8,7,7,9,9,9,7,7,8,8,8,8,9,9,8,8,9,8,9,9,9,10,10,8,9,9,10,9,8,9,8,9,8,8,8,8,8,9,10,7,9,9,8,  
8,9,7,9,8,8,9,9,6,7,8,8,6,7,8,9,9,8,8,7,9,8,8,8,8,9,8,8,8,9,8,8,9,9,8,9,7,9,9,10,10,8,9,8,8,9,9,9,9,  
9,6,7,8,8,9,10,6,6,7,8,8,7,8,9,9,8,7,8,9,8,6,7,7,9,8,8,7,9,8,8,8,8,6,6,6,7,7,7,8,9,8,8,6,7,8,8,7,8,9,  
10,8,8,8,9,8,8,8,9,8,9,8,9,8,8,8,8,8,7,7,7,8,7,8,8,9,8,8,7,7,8,8,8,8,9,8,8,8,8,9,9,8,8,9,8,8,  
8,8,9,9,10,9,9,8,8,8,7,9,9,9,8,9,9,8,8,8,9,8,9,9,9,9,9,9,9,9,9,10,9,9,8,9,8,9,9,10,6,7,8,8,8,8,9,10,  
10,7,7,9,9,10,8,8,9,9,8,8,9,9,9,8,7,9,9,10,10,9,9,9,8,8,8,8,9,9,7,8,8,9,8,8,8,9,9,8,8,8,9,8,  
8,9,9,9,8,8,9,10,8,8,9,9,10,10,9,10,8,8,8,9,5,9,10,9,6,7,8,8,8,9,8,7,7,7,8,8,7,7,9,10,8,7,7,8,8,8,8,  
9,10,10,8,8,9,8,9,9,6,7,9,9,6,7,8,9,6,7,8,9,10,8,8,8,7,7,8,7,8,8,8,8,8,8,8,8,8,7,8,9,10,9,9,9,10,10,  
8,8,9,9,10,9,8,9,10,8,9,10,8,8,8,9,10,9,7,9,8,8,8,8,8,9,10,8,8,9,8,8,8,7,10,7,9,8,9,8,9,8,8,9,8,9,9,9,  
8,9,10,9,9,9,10,9,8,8,9,10,9,9,9,10,10,10,8,9,8,9,9,9,10,9,9,7,8,8,8,8,9,8,9,9,8,8,8,9,10,9,8,9,10,8,  
8,8,9,10,10,9,7,6,7,8,9,9,9,9,10,8,8,8,7,8,9,10,9,10,10,10,8,8,8,9,9,9,9,10,10,8,9,8,9,9,8,9,10,9,9,  
9,10,9,9,10,10,7,9,9,10,9,10,8,9,10,8,9,6,7,8,9,9,7,9,10,9,8,8,7,9,10,8,9,10,10,8,8,8,9,8,9,8,8,9,9,  
10,9,10,10,10,9,9,9,9,10,9,10,10,10,8,8,8,8,8,9,9,9,8,8,8,7,8,9,10,9,9,10,10,7,8,8,9,9,10,8,9,8,  
8,9,9,8,9,9,10,8,8,8,8,8,9,9,9,8,8,8,8,9,10,6,6,7,10,8,7,7,9,8,8,8,8,9,8,8,8,9,9,9,8,8,8,9,8,9,8,  
9,10,10,8,8,9,8,9,9,9,9,9,8,9,9,8,9,8,8,8,8,8,9,10,9,9,8,8,8,9,10,9,10,9,10,8,9,9,9,8,8,9,10,  
10,9,8,8,9,10,9,9,8,8,8,8,8,9,9,10,8,8,9,8,8,7,7,9,8,9,8,9,10,8,8,8,9,10,8,7,7,9,10,9,8,9,10,9,8,8,9,  
8,9,9,9,7,9,10,9,9,8,8,8,9,8,9,7,10,10,8,8,8,9,10,9,8,9,8,8,9,9,9,9,9,10,10,10,9,10,9,9,9,10,9,9,  
6,7,9,8,9,9,9,10,9,8,8,9,9,9,7,8,8,9,10,8,9,9,10,8,8,8,9,10,9,10,9,10,9,10,9,9,8,9,9,8,9,10,10,10,9,  
8,9,9,10,10,9,10,10,10,8,9,10,9,8,8,9,9,10,8,8,8,8,9,9,9,9,9,10,8,8,8,9,9,9,8,9,10,9,7,7,7,10,10,1  
0,9,9,9,8,8,7,8,9,10,8,8,9,9,10,8,8,9,10,8,9,10,10,10,7,6,7,9,10,10,7,6,6,7,8,6,7,8,9,8,8,8,8,9,8,7,7  
9,8,9,8,8,9,8,8,8,9,9,10,8,9,10,10,8,9,9,10,10,10,9,10,9,10,8,9,9,10,10,10,9,9,10,9,10,10,8,7,7,9,  
10,9,8,9,10,10,8,8,9,10,10,8,8,9,8,9,9,10,10,10,8,9,10,9,10,10,7,7,9,10,9,8,9,9,8,8,8,9,7,10,9,9,10  
10,10,9,9,7,7,10,7,10,10,10,10,10,7,8,9,8,9,9,9,10,10,8,9,9,7,8,9,10,10,10,10,10,9,8,8,9,9,10,8,8,  
9,10,8,8,8,8,9,9,9,9,9,8,8,9,10,8,9,9,10,10,10,9,8,8,9,8,7,9,10,10,10,9,9,9,8,8,9,9,9,9,8,9,8,9,1  
0,10,10,9,9,10,9,9,9,9,10,9,9,10,10,10,6,7,9,10,8,9,8,8,9,8,8,9,9,9,10,7,9,9,9,8,6,7,9,10,6,7,10,9,9  
8,7,7,9,8,8,9,9,10,10,8,8,9,7,8,9,9,10,10,9,9,9,8,9,10,10,10,7,9,10,10,9,10,8,8,7,8,9,10,9,9,9,10,1  
0,8,9,9,9,9,8,9,10,10,10,10,8,9,8,9,10,10,7,8,9,10,10,9,9,9,10,8,8,9,10,10,9,9,9,10,8,9,9,9,9,10,10  
9,9,10,10,8,8,9,9,10,9,9,9,9,10,10,8,9,8,9,9,8,8,9,8,9,9,9,10,10,8,9,10,9,10,10,8,8,9,10,10,9,10,1  
0,10,10,8,8,9,8,9,9,10,11,10,10,10,10,10,10,8,9,11,11,10,10,8,9,9,10,9,9,10,9,10,8,8,8,9,10,10,9,  
8,9,8,9,9,9,9,10,10,10,8,8,9,10,10,9,10,10,10,9,8,8,9,10,9,10,10,10,10,8,9,9,10,8,9,9,10,10,10,10,  
9,10,10,10,7,9,9,10,10,10,8,9,10,8,9,9,8,8,8,8,9,9,9,9,8,8,8,9,9,9,9,8,8,8,9,10,8,8,9,10,8,9,8,  
9,10,8,8,9,10,8,9,9,10,9,10,10,9,10,10,8,8,8,8,9,10,10,10,8,9,9,9,8,9,10,10,9,8,8,9,9,10,9,8,9,10,8  
9,9,10,9,10,9,9,8,9,9,10,10,10,10,8,8,9,8,8,8,9,10,10,9,10,9,8,8,8,9,10,10,10,10,10,7,8,8,9,10,10,  
10,10,10,8,9,8,8,9,10,10,10,10,9,8,8,9,10,9,7,7,9,10,10,10,8,9,10,8,8,9,10,8,9,10,10,10,10,10,8,  
8,8,9,10,9,9,10,10,10,10,8,9,10,10,8,8,9,10,10,7,10,9,10,10,10,10,10,10,8,9,8,8,8,9,8,8,9,10,8,9,9  
8,9,9,8,9,9,10,9,8,8,9,10,10,8,9,10,10,10,7,7,10,9,10,10,9,10,10,8,8,9,10,10,10,10,10,8,8,9,10,10  
10,10,9,8,8,9,9,10,9,10,10,10,10,9,10,10,10,8,9,10,10,10,10,7,7,8,8,9,7,9,10,10,8,8,9,10,10,8,9,1  
0,10,9,9,10,8,9,10,10,10,10,10,10,10,8,9,9,10,10,10,10,6,7,7,9,10,10,10,9,9,8,8,8,9,8,9,11,10,  
9,8,8,8,8,9,10,11,9,9,9,9,9,10,10,8,9,9,9,9,9,8,9,8,9,9,9,9,10,9,9,9,10,9,10,10,9,9,8,9,10,7,10,10,1  
0,10,10,9,9,9,8,9,10,10,10,8,9,9,10,10,10,10,10,9,10,7,8,8,9,10,10,8,8,8,8,8,9,10,10,8,9,9,8,9,9,





```

        if(get(l,r)<=5000)f[l][r][get(l,r)]=1;//处理直接拼接的
    }
    for(int len=2;len<=13;len++)
    for(int l=1;l<=13;l++){
        int r=l+len-1;
        if(r>13)continue;
        for(int k=l;k<r;k++){
            for(int v1=1;v1<=5000;v1++)if(f[l][k][v1])
            for(int v2=1;v2<=5000;v2++)if(f[k+1][r][v2]){//由于可以任意添加括号,所以能用+,*连接两段
                if(v1+v2<=5000)f[l][r][v1+v2]=1;//相加得到的
                if(v1*v2<=5000)f[l][r][v1*v2]=1;//相乘得到的
            }
        }
    }
    for(int i=1;i<=5000;i++)ans[i]=-1;
    for(int r=1;r<=13;r++)
    for(int v=1;v<=5000;v++)
        if(f[1][r][v]&&!~ans[v])ans[v]=r;
}
void solve(){
    scanf("%d",&n);
    printf("%d\n",ans[n]);
}
int main()
{
    scanf("%d",&tes);
    while(tes--)solve();
    return 0;
}

```

## 问题 E: 同构字符串

Hdu6863:

有一个长度为N的只包含小写字母的字符串，现在问你，能否将这个字符串分成K个等长的字符串，使得每个字符串作为字符串环的时候，他们是相等的。

这题对unordered\_map好像不是很友好，我开vector来存就过了，不然就TLE。

然后，我们可以对所有的因子处理出来，然后，对于枚举长度为len的时候，只需要把第一个串的len种字符串环的哈希值都存进去就可以了，后面的几个字符串只需要直接判断就可以了。

当然，这道题还可以用最小表示法来做，如果被卡哈希了，可以试一下。

题目大意：给你一个字符串，问你是否能将这个字符串分为 k 个相同地字符串，并且这 k 个字符串同构。

同构：如果将一个字符串的前几个字符按顺序移到后面，使得这个字符串和另一个字符串相同，那么这两个字符串同构

解题思路：hash 来求解，求长度的每一个因子，然后将这个串分为若干个长度相同地子串。然后我们求得第一个字串的所有同构字串的 hash 值，判断剩下字串的 hash 值是否都包含在这个集合里即可

Hash: [https://blog.csdn.net/nka\\_kun/article/details/81254013](https://blog.csdn.net/nka_kun/article/details/81254013)

代码 1:

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int mod=1e7+7;
const int base=1331;
const int maxn=5e6+10;
int t,n;
char s[maxn];
int p[maxn],a[maxn];
int jl[mod];
inline ll read(){
    ll s=0,w=1;char ch = getchar();
    while(ch<48 || ch>57) {
        if(ch=='-') w=-1;ch = getchar();
    }
    while(ch>=48&&ch<=57) s = (s<<1) + (s<<3) + (ch^48),ch=getchar();
    return s*w;
}
int js(int l,int r)
{
    return (a[r]-1ll*a[l-1]*p[r-l+1]%mod+mod)%mod;
}
int tot=0;
bool pd(int k,int id)
{
    if(k==1) return false;
    int len=n/k;
    int l=1,r=len;
    jl[js(1,len)]=id;
    for(int i=1;i<len;i++)
    {
        int temp1=(1ll*js(i+1,len)*p[i]%mod+js(1,i))%mod;
        jl[temp1]=id;
    }
    for(int i=1;i<=k;i++)
    {
```

```

        int temp=js(l,r);
        if(jl[temp]!=id) return false;
        l+=len,r+=len;
    }
    return true;
}
int main()
{
    int t;
    t=read();
    p[0]=1;
    for(int i=1;i<maxn;i++) p[i]=(1ll*p[i-1]*base)%mod;
    while(t--)
    {
        n=read();
        scanf("%s",s+1);
        for(int i=1;i<=n;i++) a[i]=(1ll*a[i-1]*base%mod+(s[i]-'a'+1))%mod;
        int flag=0;
        for(int i=1;1ll*i*i<=1ll*n&&!flag;i++)
        {
            if(n%i==0)
            {
                flag|=pd(i,tot++);
                flag|=pd(n/i,tot++);
            }
        }
        if(!flag) cout<<"No"<<endl;
        else cout<<"Yes"<<endl;
    }
    return 0;
}

```

代码 2:

```

#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int M=1e7+5;
int n,st[M];
char s[M],t[M];
void play(int x,int y,int len){
    for(int i=1,j=x;i<=len;i++,j++)t[i]=s[j];
    for(int i=1;i<=len;i++)t[i+len]=t[i];
    int i=1,j=2,k;
    while(i<=len&&j<=len){
        for(k=0;k<len&&t[i+k]==t[j+k];k++);
    }
}

```

```

        if(k==len)break;
        if(t[i+k]>t[j+k]){
            i=i+k+1;
            if(i==j)i++;
        }
        else{
            j=j+k+1;
            if(j==i)j++;
        }
    }
    st[x]=min(x+i-1,x+j-1);
}
int chk(int x1,int y1,int s1,int x2,int y2,int s2,int len){
    for(int i=0;i<len;i++){
        if(s[st[x1]+i>y1?st[x1]+i-len:st[x1]+i]!=s[st[x2]+i>y2?st[x2]+i-len:st[x2]+i])return 0;
    }
    return 1;
}
int work(int len){
    play(1,len,len);
    for(int i=len+1;i<=n;i+=len){
        play(i,i+len-1,len);
        if(!chk(1,len,st[1],i,i+len-1,st[i],len))return 0;
    }
    return 1;
}
void solve(){
    scanf("%d",&n);
    scanf("%s",s+1);
    if(n==1){puts("No");return;}
    for(int len=1;len*len<=n;len++){if(n%len==0){
        if(work(len)){puts("Yes");return;}
        if(len>1&&len*len!=n)if(work(n/len)){puts("Yes");return;}
    }
    puts("No");
}
int main()
{
    int tes;scanf("%d",&tes);
    while(tes--)solve();
    return 0;
}

```

问题 F: 金矿探险

/\*

问题求最大距离最小值，容易想到二分答案，将求解变成判定，会降低难度。

对于二分的答案  $lim$ ，要求树中任意两条链相加长度  $\leq lim$ ，否则答案  $lim$  不成立。

可以做类似求树的直径的 dp，在树的直径 dp 中， $d[x]$  表示  $x$  为子树根，子树内最远点到  $x$  距离。

这里可以引入  $f[x][i]$  表示  $x$  为子树根，子树内有  $i$  条  $a$  边，最远点到  $x$  距离。

在求树的直径中， $ans$  可能为  $d[x]+d[y]+e(x,y)$  ( $y$  为  $x$  的儿子)，或者某个子树内的  $ans$ ，但某个子树内的  $ans$  也是由  $d[x]+d[y]+e(x,y)$  构成，只要限定两条链相加超过  $lim$  不能转移，就能保证  $x$  子树内不存在  $>lim$  的路径，要超过  $lim$ ，未来只需要考虑  $x$  最长链和  $x$  兄弟最长链的合并。

转移有：

$g[x]=\min(g[x],\max(f[x][j],f[y][k]+ea(x,y)))$  ( $x \rightarrow y$  用  $a$  边)

$g[x]=\min(g[x],\max(f[x][j],f[y][k]+eb(x,y)))$  ( $x \rightarrow y$  用  $b$  边)

直接这样会影响本层状态， $f[x][0]$  可能影响  $f[x][1]$  等，需要倒着枚举第二维(类似背包)，

用一个临时数组存储，用  $g[i]$  暂存更新的  $f[x][i]$ ，等转移完，再将  $f[x][i]=g[i]$ 。

最后，如果  $f[1][K]$  能转移到，就说明答案  $lim$  成立。单组数据时间复杂度为  $n*k^2*\log(len)$

\*/

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
typedef long long ll;
```

```
const int M=20005;
```

```
int n,K,sz[M]; //sz[x]表示 x 子树内边的数量，超过 K 就赋值为 K
```

```
ll lim,f[M][25];
```

```
//f[x][i]表示 x 为子树根，子树内有 i 条 a 边，最远点到 x 的距离
```

```
int tot,head[M],Next[M*2],vet[M*2],ea[M*2],eb[M*2];
```

```
void add(int a,int b,int c,int d){
```

```
    Next[++tot]=head[a],vet[tot]=b,ea[tot]=c,eb[tot]=d;
```

```
    head[a]=tot;
```

```
}
```

```
void dfs(int x,int pre){
```

```
    f[x][0]=0; sz[x]=0;
```

```
    for(int i=head[x]; i; i=Next[i]){
```

```
        int y=vet[i],a=ea[i],b=eb[i];
```

```
        if(y==pre) continue;
```

```
        dfs(y,x);
```

```
        ll g[25];
```

```
        for(int j=0; j<=min(K,sz[x]+sz[y]+1); j++) g[j]=lim+1; //初始化无穷
```

```
        for(int j=0; j<=sz[x]; j++)
```

```
            for(int k=0; k<=min(sz[y],K-j); k++){
```

```
                if(f[x][j]+f[y][k]+a<=lim)
```

```
                    g[j+k+1]=min(g[j+k+1],max(f[x][j],f[y][k]+a));
```

```
                if(f[x][j]+f[y][k]+b<=lim)
```

```
                    g[j+k]=min(g[j+k],max(f[x][j],f[y][k]+b));
```

```

        }
        sz[x]=min(sz[x]+sz[y]+1,K);
        for(int j=0; j<=sz[x]; j++) f[x][j]=g[j];
    }
}

void solve(){
    scanf("%d%d",&n,&K);
    tot=0; for(int i=1; i<=n; i++) head[i]=0;
    ll l=1,r=0,ans=-1;
    for(int i=2; i<=n; i++){
        int x,y,a,b; scanf("%d%d%d%d",&x,&y,&a,&b);
        add(x,y,a,b); add(y,x,a,b);
        r+=max(a,b);
    }
    while(l<=r){
        ll mid=(l+r)/2;
        lim=mid;
        dfs(1,0);
        if(f[1][K]!=lim+1){
            ans=mid; r=mid-1;
        }else l=mid+1;
    }
    printf("%lld\n",ans);
}

int main(){
    int T; scanf("%d",&T);
    while(T--) solve();
    return 0;
}

```