分块

分块?

常数小，能够**在线**，思维难度小。

例题1

P3203 [HNOI2010]弹飞绵羊

容易发现是一棵树/森林，因为一个点只能往后面连边，并且只连一条边。

题目变成改变一条边的终点，维护一个点的深度。

如果你这样想就是LCT的裸题了。

但是为了维护尊严，我们不写LCT。

考虑分块，假设我们现在有一个点x，预处理跳几步能够跳出这个点所在的块，步数记为$dep[x]$，跳到的那个点记为$to[x]$（n及以后的点编号为0，看作一个不同的块）

注意到我们更新一个点要更新这个点所在的块中，编号在这个点前面的所有点，因为前面的点也可能跳到这个点，导致dep,to变化。

当然你也可以更新整个块，只是常数大了一点。

不开O2会被卡一个点，但是开了O2跑得极快。

$O(m\sqrt{n})$，然而$n\le 200000,m\le 100000$，约为5×10^7，说明常数还是挺小的。

```
#include <bits/stdc++.h>
#define MAXN 200005
using namespace std;
inline int read(){
    int x=0,f=1;
    char ch=getchar();
    while (ch<'0'||ch>'9'){
        if (ch=='-') f=-1;
        ch=getchar();
    }
    while (ch>='0'&&ch<='9'){
        x=(x<<3)+(x<<1)+(ch^'0');
        ch=getchar();
    }
    return x*f;
}
int dep[MAXN],to[MAXN];//预处理一个点跳多少步能跳出这个块，跳到哪里
int n,Size,a[MAXN],pos[MAXN];
void Rebuild(int p,int id){
    int l=(id-1)*Size,r=min(p,id*Size-1);
    for (int i=r;i>=l;--i){//倒着搞
        if (pos[a[i]]!=pos[i]) dep[i]=1,to[i]=a[i];
        else dep[i]=dep[a[i]]+1,to[i]=to[a[i]];
    }
}
int Query(int x){
    int ans=0;
    while (true){
        ans+=dep[x],x=to[x];
        if (pos[x]==0) return ans;
    }
    return -1;
}
int main(){
    n=read();
    for (int i=0;i<n;++i) a[i]=i+read();//弹到哪个位置
    Size=sqrt(n);
```

```
    for (int i=0;i<n;++i) pos[i]=i/Size+1;
    for (int i=1;i<=pos[n-1];++i) Rebuild(0x7fffffff,i);
    int m=read();
    while (m--){
        int opr=read(),i=read();
        if (opr==1){
            printf("%d\n",Query(i));
        }
        else {
            a[i]=i+read();
            Rebuild(i,pos[i]);
        }
    }
}
```

例题2

CODECHEF Nov. Challenge 2014 Chef & Churu

首先想到改变一个元素，会影响哪些函数值。

容易发现，当$L_i \leq x \leq R_i$时，

考虑分块，对于连续$\sqrt{n}$个的函数分成一块。