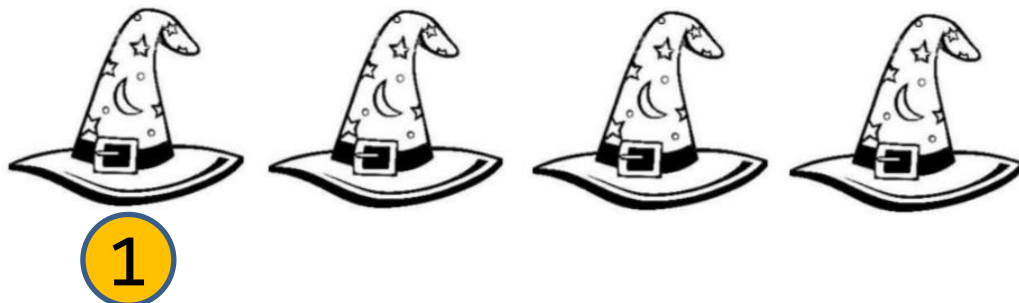


C++算法

快快编程1061

快快编程
kkcoding.net

输入
4 4
1 2 3 2
输出
4



第1次交换后



第2次交换后



第3次交换后



第4次交换后



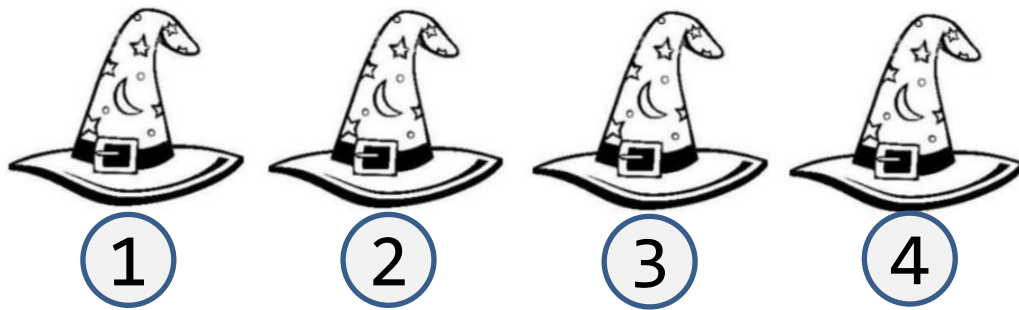
p表示1号小球此时在哪个位置

```
8      cin>>n>>m;
9      int ans=1;
10     for(int k=1;k<=m;k++){
11         cin>>p
12         if(p==ans)
13             ans=p+1;
14         else if( )
15             ;
16     }
17     cout<<ans<<endl;
```

快快编程1062

快快编程
kkcoding.net

输入
4 4
1 2 3 2
输出
4 1 3 2



第1次
交换后



第2次
交换后



$x[i]$ 表示当前第 i 个位置放有原来几号小球
 $y[i]$ 表示原来第 i 个小球当前在几号位置

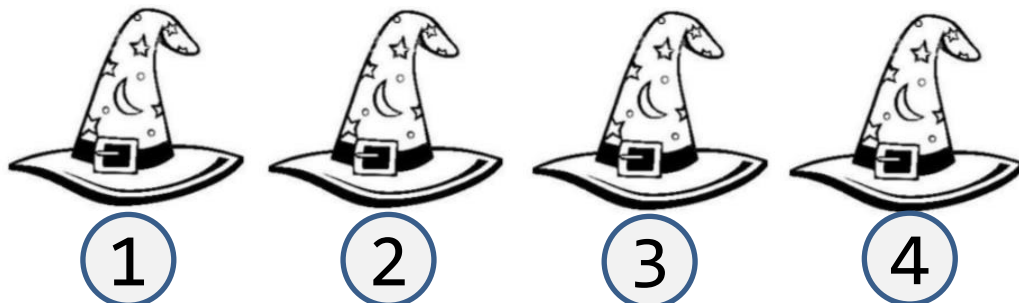
	$i=1$	$i=2$	$i=3$	$i=4$
$x[i]=$	2	3	1	4
$y[i]=$	3	1	2	4

```
8  cin>>n>>m;
9  for(int i=1;i<=n;i++)y[i]=x[i]=i;
10 for(int k=1;k<=m;k++){
11     cin>>p;
12     swap(x[p],);
13     swap(,);
14 }
```

快快编程1063

快快编程
kkcoding.net

输入
4 4
1
2
3
2
2
2 3
3 1
输出
1
3



第1次
交换后



第2次
交换后



第3次
交换后



第4次
交换后



询问: 2号球在第3次交换后在哪里?
询问: 3号球在第1次交换后在哪里?

暴力模拟

对于 q 个询问中的每个问题：
原来 k 号小球在第 j 次交换后
都从头模拟一遍前 j 次交换

```
5 void input(){
6     cin>>n>>m;
7     for(int j=1;j<=m;++j)cin>>p[j];
8     cin>>q;
9     for(int i=1;i<=q;++i)cin>>K[i]>>J[i];
10 }
```

$p[j]$ 代表第 j 次交换 $p[j]$ 和 $p[j]+1$ 两个位置的球

i 号问询：原来 $K[i]$ 号球在第 $J[i]$ 次交换后的位置

```
23 int main(){
24     freopen("hattrick3.in","r",stdin);
25     freopen("hattrick3.out","w",stdout);
26     input();
27     solveBF();
28     return 0;
29 }
```

暴力求解
brute force

老师
检查

完成
程序

暴力
模拟

```
5 void input(){
6     cin>>n>>m;
7     for(int j=1;j<=m;++j)cin>>p[j];
8     cin>>q;
9     for(int i=1;i<=q;++i)cin>>K[i]>>J[i];
10 }
```

pos记录原K[i]号球此时的位置

```
11 void solveBF(){
12     for(int i=1;i<=q;++i){
13         int pos=K[i];
14         for(int j=1;j<=J[i];++j)
15             if(p[j]==pos)
16                 pos=p[j]+1;
17             else if(p[j]+1==pos)
18                 pos=p[j];
19         ans[i]=pos;
20     }
21     for(int i=1;i<=q;++i)cout<<ans[i]<<endl;
22 }
```

复杂度 $O(qm)$

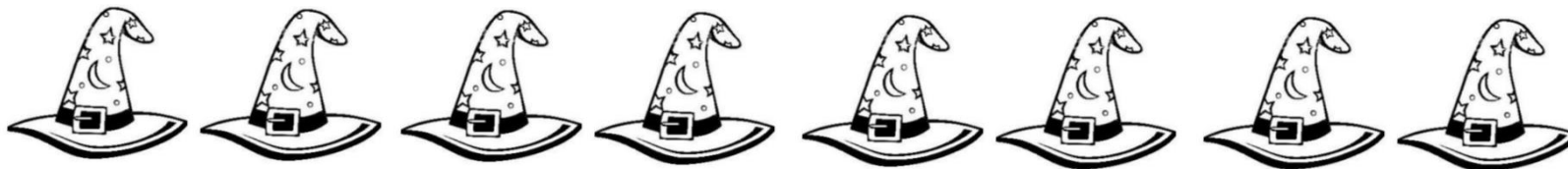
离线询问

给定序列1到n，依次有m次邻居交换，q次询问
原来数字k在第j次交换后在哪里

离线
询问

看完所有问题，挑最简单的答题顺序

按时间自然顺序处理询问
按照事件先后
模拟交换+回答问题



```

11 bool cmp(const int&a,const int&b){
12     return J[a]<J[b]||J[a]==J[b]&&K[a]<K[b];
13 }

```

询问里问到的交换次数第*i*小的
是原第id[i]个询问

```

14 void solve(){
15     for(int i=1;i<=q;++i)id[i]=i;
16     sort(id+1,id+1+q,cmp);

```

x[i]目前第i个原来几号

y[i]原来第i个现在几号

```

17     for(int i=1;i<=n;++i)y[i]=x[i]=i;
18     for(int k=1,j=1;k<=q;++k){
19         for(;j<=J[id[k]];++j){
20             swap(x[p[j]],x[p[j]+1]);
21             swap(y[x[p[j]]],y[x[p[j]+1]]);
22         }
23         ans[id[k]]=y[K[id[k]]];
24     }
25     for(int i=1;i<=q;++i)cout<<ans[i]<<endl;
26 }

```

老师
检查

完成
程序

小结

离线问询

不需要即时回答

看完所有问题，挑最简答题顺序

思考：哪类答题顺序可能较容易？

被其他问题依赖的基础问题
为其他问题提供有用信息

正难则反

潜在顺序

最值

边角

对拍

快快编程
kkcoding.net

对拍

为了验证程序的正确性

同一个问题
用2种解法求解
查看输出结果是否一致

使用暴力解法做对比

使用随机生成大型数据

对拍是高手必备技能

随机数据

```
5 void input(){
6     cin>>n>>m;
7     for(int j=1;j<=m;++j)cin>>p[j];
8     cin>>q;
9     for(int i=1;i<=q;++i)cin>>K[i]>>J[i];
10 }
11 void genData(){
12     n=100; m=200;
13     for(int j=1;j<=m;++j)
14         p[j]=rand()%(n-1)+1;
15     q=50;
16     for(int i=1;i<=q;++i){
17         K[i]=rand()%n+1;
18         J[i]=rand()%m+1;
19     }
20 }
```

随机范围1到n-1

随机范围1到n

随机范围1到m

对拍模式用于测试 2种解法2种输出

```
49 int main(){
50     freopen("hat.in", "r", stdin);
51     freopen("hat.out", "w", stdout);
52     // input();
53     genData();
54     solve();
55     freopen("hatBF.out", "w", stdout);
56     solveBF();
57     return 0;
58 }
```

hat.out		hatBF.out	
1	10	1	10
2	35	2	35
3	1	3	1
4	17	4	17
5	91	5	91
6	25	6	25
7	45	7	45
8	88	8	88
9	97	9	97
10	98	10	98
11	26	11	26
12	14	12	14
13	49	13	49
14	59	14	59
15	83	15	83
16	91	16	91
17	90	17	90
18	8	18	8
19	73	19	73
20	42	20	42
21	42	21	42
22	93	22	93
23	62	23	62
24	15	24	15
25	57	25	57
26	4	26	4
27	73	27	73
28	29	28	29
29	60	29	60
30	77	30	77
31	70	31	70
32	45	32	45
33	82	33	82
34	23	34	23
35	34	35	34

Line: 50 Col: 3 Sel: 0

正解模式用于提交 1种解法1种输出

```
49 int main(){
50     freopen("hat.in", "r", stdin);
51     freopen("hat.out", "w", stdout);
52     input();
53     // genData();
54     solve();
55     // freopen("hatBF.out", "w", stdout);
56     // solveBF();
57     return 0;
58 }
```

骗分模式用于提交 1种暴力解法1种输出

```
49 int main(){
50     freopen("hat.in", "r", stdin);
51     freopen("hat.out", "w", stdout);
52     input();
53     // genData();
54     // solve();
55     // freopen("hatBF.out", "w", stdout);
56     solveBF();
57     return 0;
58 }
```

写作 顺序

1

4

3

5

2

```
49 int main(){
50     freopen("hat.in", "r", stdin);
51     freopen("hat.out", "w", stdout);
52     input();
53     genData();
54     solve();
55     freopen("hatBF.out", "w", stdout);
56     solveBF();
57     return 0;
58 }
```

快快编程作业

1061

1062

1063