

## T433080 快速排序的败北

### 算法1

`next_permutation()` 打表  $n = 10$  的情况, 就能知道最劣情况长什么样子,  $n = 1000$  用类似的形式填。

例如  $[1, 2, \dots, 499, 1000, 500, 501, \dots, 999]$  这个就可以。

### 算法2

假设已经构造好了  $i$  个数的解 (假设为序列  $a$ ), 那么对于  $i + 1$  个数的问题, 只需将最大值或者最小值插入  $a$  的中间就能构造出  $i + 1$  个数的解  $a'$ , 递归即可。

```
1 void f(int l, int r){
2     if(l >= r) return;
3     f(l + 1, r);
4     swap(a[l], a[(l+r)/2]);
5 }
```

## U86432 捞鱼(fish)

记  $mp[x][y]$  代表以  $(x, y)$  为左上角的渔网能捕到多少条鱼, 使用 `map<pair<int,int>, int>` `mp`; 维护这个信息。

观察到边长  $r$  很小, 可以  $r^2$  暴力把一条鱼的贡献打进 `map`。修改 `map` 的同时可以快速维护答案数组  $ans$  (注意  $ans[0]$  要赋初值)。

复杂度  $O(kr^2 \log k)$ 。

```
1 void solve(){
2     ans[0]=(long long)(n-r+1)*(m-r+1);
3     for(int i=1,x,y;i<=k;i++){
4         cin>>x>>y;
5         for(int i=max(1,x-r+1);i<=min(x,n-r+1);i++){
6             for(int j=max(1,y-r+1);j<=min(y,m-r+1);j++){
7                 pair<int,int> cur={i,j};
8                 ans[mp[cur]]--;
9                 ans[mp[cur]+1]++;
10                mp[cur]++;
11            }
12        }
13    }
14    for(int i=0;i<=k;i++) cout<<ans[i]<<'\n';
15 }
```

## U184510 冒泡排序趟数期望

对于排列  $p$ :

- $pos[x]$  代表  $x$  的位置, 即  $p[pos[x]] = x$
- $inv[x] = x$  之前  $> x$  的数字数量, 即  $inv[x] = \sum_{i=1}^{pos[x]} [p[i] > x]$

有:

- $inv[x] \in [0, n - x]$
- 不同的  $inv[x]$  数组恰好有  $n!$  个, 和所有排列一一对应 (可以从后往前构造)

先把所求差分  $cnt[=k] = cnt[\leq k] - cnt[\leq k - 1]$

问题转化为求  $max(inv[x]) \leq k$  的方案数, 对于  $x \geq n - k$  随便填 (共  $(k + 1)!$  种填法),  $x < n - k$  限制  $inv[x] \leq k$  (共  $(n + 1)^{n-k-1}$  种填法)。方案数为:

- $cnt[\leq k] = (k + 1)!(k + 1)^{n-k-1} = k!(k + 1)^{n-k}$

复杂度  $O(n \log n)$ 。

```
1 int ans = 0;
2 for(int k=0;k<=n;k++){
3     cnt[k] = fac[k] * qpow(k+1, n-k) % P;
4     ans = (ans + k * (cnt[k] - cnt[k-1] + P) % P) % P;
5 }
6 ans = ans * qpow(fac[n], P-2) % P;
7 cout<<ans<<endl;
```

## P6852 Mex

参考 <https://www.luogu.com.cn/article/qlxec4ah>

对于每个值将所有  $(l, r, v)$  取交集为  $ok[v]$ , 取并集为  $ban[v]$ , 注意交集/并集不是连续段则直接无解 (0 除外)。

对于一个值  $v$ , 可以放的位置就是  $AND_{x>v} \{ok[x]\} - ban[v]$ , 用线段树或者并查集在可行位置里随便放一个即可。

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 #define MAXN 500005
4
5 pair<int,int> OR(pair<int,int> x, pair<int,int> y){
6     return {min(x.first,y.first), max(x.second,y.second)};
7 }
8
```

```

9 pair<int,int> AND(pair<int,int> x, pair<int,int> y){
10     return {max(x.first,y.first), min(x.second,y.second)};
11 }
12
13 int n,m;
14 vector<pair<int,int> > adj[MAXN];
15 pair<int,int> ok[MAXN], ban[MAXN];
16 int vis[MAXN];
17 int ans[MAXN];
18
19 int fa[MAXN];
20 int findr(int x) { return (fa[x]==x)?x:fa[x]=findr(fa[x]);}
21 bool find(int l, int r, int v){//找[l,r]区间中第一个空位放v
22     //if(l > r) return 0;
23     int x = findr(l);
24     if(x <= r){
25         ans[x] = v;
26         fa[x] = x+1;
27         return 1;
28     }
29     return 0;
30 }
31
32 int main(){
33     cin>>n>>m;
34     for(int i=1,l,r,v;i<=m;i++){
35         cin>>l>>r>>v;
36         adj[v].push_back({l,r});
37         if(v == n+1 && r-l+1 < n+1){
38             cout<<"-1";
39             return 0;
40         }
41     }
42
43     ok[n+1] = {0,n};
44     for(int v=n;v>=1;--v){
45         ok[v] = ok[v+1], ban[v] = {n+1,-1};
46         for(auto p:adj[v]){
47             ok[v] = AND(ok[v], p);
48             ban[v] = OR(ban[v], p);
49         }
50     }
51
52     //fa
53     for(int i=0;i<=n+1;i++) fa[i] = i;
54     //放0要特殊处理
55     for(auto p:adj[0]){
56         vis[p.first]++;
57         vis[p.second+1]--;
58     }
59
60     bool flag = 0;

```

```

61     for(int i=0;i<=n;++i){
62         if(i>0) vis[i] += vis[i-1];
63         if(vis[i]) continue;
64         if(ok[1].first<=i && i<=ok[1].second){
65             ans[i] = 0;
66             fa[i] = i+1;
67             flag = 1;
68             break;
69         }
70     }
71
72     if(flag == 0){
73         cout<<"-1";
74         return 0;
75     }
76     //放v=1~n
77     for(int v=1;v<=n;++v){
78         int l1 = ok[v+1].first, r1 = min(ok[v+1].second,
ban[v].first-1);
79         if(find(l1, r1, v)) continue;
80         int l2 = max(ok[v+1].first, ban[v].second+1), r2 =
ok[v+1].second;
81         if(find(l2, r2, v)) continue;
82         cout<<"-1";
83         return 0;
84     }
85     for(int i=0;i<=n;++i) cout<<ans[i]<<" ";
86     return 0;
87 }

```