

原创 高级数据结构模板

2019-10-17 10:52:57 _Y_-Y_ 阅读数 6 文章标签: ACM 更多

编辑

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_44410512/article/details/102601427

线段树

```

1  /*
2  敌兵布阵
3  HDU - 1166
4  https://cn.vjudge.net/problem/HDU-1166
5  题面：线段树裸题
6  输入： T组数据 N个工兵营地 N个第i个开始有多少人
7          有四种命令 到end结束
8          Add i j, i和j为正整数，表示第i个营地增加j个人（j不超过30）
9          Sub i j, i和j为正整数，表示第i个营地减少j个人（j不超过30）；
10         Query i j, i和j为正整数，i<=j，表示询问第i到第j个营地的总人数；
11  样例输入：
12         1
13         10
14         1 2 3 4 5 6 7 8 9 10
15         Query 1 3
16         Add 3 6
17         Query 2 7
18         Sub 10 2
19         Add 6 3
20         Query 3 10
21         End
22  样例输出：
23         Case 1:
24         6
25         33
26         59
27  */
28  #include <cstdio>
29  #include <cstring>
30  #include <iostream>
31  #include <algorithm>
32  using namespace std;
33  #define maxn 100000
34  #define ll long long
35  ll tree[maxn<<2];
36  ll num[maxn];
37  char c[10];
38  int n;
39  void update(int now){
40     tree[now]=tree[now<<1]+tree[now<<1|1];
41 }
42 void build(int l, int r, int now){
43     if(l==r){
44         tree[now]=num[l];
45         return ;
46     }
47     int mid=(l+r)/2;
48     build(l, mid, now<<1);
49     build(mid+1, r, now<<1|1);
50     update(now);
51 }
52 void change(int l, int r, int now, int x, ll c){
53     if(l==r){
54         tree[now]+=c;
55         return;
56     }
57     int mid=(l+r)/2;
58     if(x<=mid) change(l, mid, now<<1, x, c);
59     else change(mid+1, r, now<<1|1, x, c);

```

编程语言大PK，你选谁？

关闭

```

60     update(now);
61 }
62 ll seartch(int l, int r, int now, int i, int j){
63     if(i<=l&&r<=j) return tree[now];
64     ll ans=0;
65     int mid=(l+r)/2;
66     if(i<=mid) ans+=seartch(l, mid, now<<1, i, j);
67     if(j>mid) ans+=seartch(mid+1, r, now<<1|1, i, j);
68     return ans;
69 }
70 int main(){
71     int T;
72     scanf("%d", &T);
73     for(int s=1;s<=T;s++){
74         printf("Case %d:\n", s);
75         scanf("%d", &n);
76         for(int i=1;i<=n;i++){
77             scanf("%lld", &num[i]);
78         }
79         build(1, n, 1);
80         ll a, b;
81         while(~scanf("%s", c)){
82             if(c[0]=='E') break;
83             scanf("%lld %lld", &a, &b);
84             if(c[0]=='A') change(1, n, 1, (int)a, b);
85             if(c[0]=='S') change(1, n, 1, (int)a, -b);
86             if(c[0]=='Q') printf("%lld\n", seartch(1, n, 1, (int)a, (int)b));
87         }
88     }
89     return 0;
90 }

```

线段树+lazy

```

1  /*
2  Just a Hook
3  HDU - 1698
4  https://cn.vjudge.net/problem/HDU-1698#author=0
5  题面：开始都是1，每次将一段区间内的所有数字都变为c
6  样例输入：
7      1
8      10
9      2
10     1 5 2
11     5 9 3
12 样例输出：
13     24
14 解法：线段树+Lazy
15  */
16 #include <cstdio>
17 #include <cstring>
18 #include <iostream>
19 #include <algorithm>
20 using namespace std;
21 #define maxn 500000
22 #define ll long long
23 ll tree[maxn<<2];
24 ll lazy[maxn<<2];
25 int n,m;
26 void update(int now){
27     tree[now]=tree[now<<1]+tree[now<<1|1];
28 }
29 void update_lazy(int now,int d){
30     if(lazy[now]!=0){
31         tree[now<<1]=lazy[now]*(d-d/2);
32         tree[now<<1|1]=lazy[now]*(d/2);
33         lazy[now<<1]=lazy[now];
34         lazy[now<<1|1]=lazy[now];
35         lazy[now]=0;
36     }

```

```

37 }
38 void build(int l, int r, int now){
39     if(l==r){
40         tree[now]=1;
41         return ;
42     }
43     int mid=(l+r)/2;
44     build(l, mid, now<<1);
45     build(mid+1, r, now<<1|1);
46     update(now);
47 }
48 void change_lazy(int l, int r, int now, int i, int j, ll c){
49     if(i<=l&&r<=j){
50         tree[now]=c*(r-l+1);
51         lazy[now]=c;
52         return;
53     }
54     update_lazy(now,(r-l+1));
55     int mid=(l+r)/2;
56     if(i<=mid) change_lazy(l, mid, now<<1, i, j, c);
57     if(j>mid) change_lazy(mid+1, r, now<<1|1, i, j, c);
58     update(now);
59 }
60 ll seartch(int l, int r, int now, int i, int j){
61     if(i<=l&&r<=j) return tree[now];
62     update_lazy(now,r-l+1);
63     ll ans=0;
64     int mid=(l+r)/2;
65     if(i<=mid) ans+=seartch(l, mid, now<<1, i, j);
66     if(j>mid) ans+=seartch(mid+1, r, now<<1|1, i, j);
67     return ans;
68 }
69 int main(){
70     int T,t=1;
71     scanf("%d", &T);
72     while(T--){
73         memset(tree,0,sizeof tree);
74         memset(lazy,0,sizeof lazy);
75         scanf("%d %d", &n, &m);
76         build(1, n, 1);
77         ll a, b, d;
78         while(m--){
79             scanf("%lld %lld %lld", &a, &b, &d);
80             change_lazy(1, n, 1, (int)a, (int)b, d);
81         }
82         printf("Case %d: The total value of the hook is %lld.\n", t++, seartch(1, n, 1, 1, n));
83     }
84     return 0;
85 }
86 #include <bits/stdc++.h>
87 using namespace std;
88 #define maxn 510000
89 struct node{
90     int lazy;
91     int num;
92 }tree[maxn];
93 void update(int now,int d){
94     if(tree[now].lazy!=0){
95         tree[now<<1].num=tree[now].lazy*(d-d/2);
96         tree[now<<1|1].num=tree[now].lazy*(d/2);
97         tree[now<<1].lazy=tree[now].lazy;
98         tree[now<<1|1].lazy=tree[now].lazy;
99         tree[now].lazy=0;
100     }
101 }
102 void build(int now,int l,int r){
103     tree[now].lazy=0;
104     if(l==r){
105         tree[now].num=1;
106         return ;
107     }

```

```

108     int mid=(l+r)>>1;
109     build(now<<1,l,mid);
110     build(now<<1|1,mid+1,r);
111     tree[now].num=tree[now<<1].num+tree[now<<1|1].num;
112 }
113 void change(int now,int l,int r,int i,int j,int c){
114     if(i<=l&&r<=j){
115         tree[now].num=c*(r-l+1);
116         tree[now].lazy=c;
117         return ;
118     }
119     update(now,r-l+1);
120     int mid=(l+r)>>1;
121     if(i<=mid) change(now<<1,l,mid,i,j,c);
122     if(j>mid) change(now<<1|1,mid+1,r,i,j,c);
123     tree[now].num=tree[now<<1].num+tree[now<<1|1].num;
124 }
125 int seartch(int now,int l,int r,int i,int j){
126     if(i<=l&&r<=j){
127         return tree[now].num;
128     }
129     int ans=0;
130     update(now,r-l+1);
131     int mid=(l+r)>>1;
132     if(i<=mid) ans+=seartch(now<<1,l,mid,i,j);
133     if(j>mid) ans+=seartch(now<<1|1,mid+1,r,i,j);
134     return ans;
135 }
136 int main(){
137     int T,t=1;
138     int l,r,k;
139     int n,m;
140     scanf("%d",&T);
141     while(T--){
142         scanf("%d %d",&n,&m);
143         build(1,1,n);
144         while(m--){
145             scanf("%d %d %d",&l,&r,&k);
146             change(1,1,n,l,r,k);
147         }
148         printf("Case %d: The total value of the hook is %d.\n", t++, seartch(1,1,n,1,n));
149     }
150     return 0;
151 }

```

线段树+离散化

```

1  /*
2  Mayor's posters
3  POJ - 2528
4  https://cn.vjudge.net/problem/POJ-2528#author=_PI_
5  题意: 贴海报, 会覆盖掉之前的点
6  输入: T组样例 贴海报的次数 海报范围
7  样例输入:
8          1
9          5
10         1 4
11         2 6
12         8 10
13         3 4
14         7 10
15 样例输出:
16         4
17 解法:
18 离散化+区间修改
19 */
20 #include <cstring>
21 #include <cstdio>
22 #include <iostream>
23 #include <algorithm>

```

```

24 using namespace std;
25 #define maxn 100000
26 struct node {
27     int l;
28     int r;
29 } num[maxn];
30 int tree[maxn<<2];
31 int color[maxn<<1];
32 int has[10000010];
33 bool vis[maxn<<1];
34 int cnt;
35 void init(){
36     memset(tree, 0, sizeof tree);
37     memset(vis, false, sizeof vis);
38     cnt = 0;
39 }
40 void updown(int now){
41     if(tree[now] != -1){
42         tree[now << 1] = tree[now << 1 | 1] = tree[now];
43         tree[now] = -1;
44     }
45 }
46 void change(int l, int r, int now, int i, int j, int c){
47     if(i <= l && r <= j){
48         tree[now] = c;
49         return ;
50     }
51     updown(now);
52     int mid = (l + r) / 2;
53     if(i <= mid) change(l, mid, now<<1, i, j, c);
54     if(j > mid) change(mid+1, r, now<<1|1, i, j, c);
55 }
56 void seartch(int l, int r, int now, int i, int j){
57     if(tree[now] != -1){
58         vis[tree[now]] = true;
59         return;
60     }
61     int mid = (l + r) / 2;
62     if(i <= mid) seartch(l, mid, now<<1, i, j);
63     if(j > mid) seartch(mid+1, r, now<<1|1, i, j);
64 }
65 int main() {
66     int T, n;
67     scanf("%d", &T);
68     while(T--){
69         init();
70         scanf("%d", &n);
71         for(int i = 0; i < n; i++) {
72             scanf("%d %d", &num[i].l, &num[i].r);
73             color[cnt++] = num[i].l;
74             color[cnt++] = num[i].r;
75         }
76         sort(color, color + cnt);
77         cnt = unique(color, color + cnt) - color;
78         for(int i = 1; i <= cnt; i++) {
79             has[color[i - 1]] = i;
80         }
81         for(int i = 1; i <= n; i++) {
82             change(1, cnt, 1, has[num[i - 1].l], has[num[i - 1].r], i);
83         }
84         seartch(1, cnt, 1, 1, cnt);
85         int ans = 0;
86         for(int i = 1; i <= n; i++) {
87             if(vis[i]){
88                 ans++;
89                 //cout<<i<<endl;
90             }
91         }
92         printf("%d\n", ans);
93     }
94 }

```

```

95     return 0;
96 }
97
98
99
100 /*
101 Mayor's posters
102 POJ - 2528
103 https://cn.vjudge.net/problem/POJ-2528#author=_PI_
104 题意： 贴海报，会覆盖掉之前的点
105 输入： T组样例 贴海报的次数 海报范围
106 样例输入：
107         1
108         5
109         1 4
110         2 6
111         8 10
112         3 4
113         7 10
114 样例输出：
115         4
116 解法：
117 离散化+区间修改
118 */
119 #include <iostream>
120 #include <cstring>
121 #include <cstdio>
122 #include <algorithm>
123 #define maxn 100000
124 using namespace std;
125 struct TREE{
126     int l,r;
127     bool v;
128 }tree[maxn<<2];
129 struct node{
130     int l,r;
131 }num[maxn];
132 int Hash[1000010];
133 int color[maxn<<1];
134 int n;
135 void build(int l,int r,int now){
136     tree[now].l=l;
137     tree[now].r=r;
138     tree[now].v=false;
139     if(l==r) return ;
140     int mid=(l+r)>>1;
141     build(l,mid,now<<1);
142     build(mid+1,r,now<<1|1);
143 }
144 bool update(int l,int r,int now){
145     if(tree[now].v) return false;
146     if(tree[now].l==l&&tree[now].r==r){
147         tree[now].v=true;
148         return true;
149     }
150     int temp;
151     int mid=(tree[now].l+tree[now].r)>>1;
152     if(r<=mid) temp=update(l,r,now<<1);
153     else if(l>mid) temp=update(l,r,now<<1|1);
154     else{
155         bool t1=update(l,mid,now<<1);
156         bool t2=update(mid+1,r,now<<1|1);
157         temp=t1||t2;
158     }
159     tree[now].v=tree[now<<1].v&&tree[now<<1|1].v;
160     return temp;
161 }
162 void init(){
163     int cnt=0;
164     scanf("%d", &n);
165     for(int i=0;i<n;i++){

```

```

166     scanf("%d %d", &num[i].l, &num[i].r);
167     color[cnt++] = num[i].l;
168     color[cnt++] = num[i].r;
169 }
170 sort(color, color + cnt);
171 cnt = unique(color, color + cnt) - color;
172
173 build(1, cnt, 1);
174 int ans = 0;
175 for(int i = n - 1; i >= 0; i--){
176     int l = num[i].l, r = num[i].r;
177     if(update(Hash[l], Hash[r], 1)) ans++;
178 }
179 printf("%d\n", ans);
180 }
181 int main(){
182     int T;
183     scanf("%d", &T);
184     while(T--){
185         init();
186     }
187     return 0;
188 }

```

可持久化线段树

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define maxn 1000005
4  int n, m, tot;
5  int a[maxn], root[maxn];
6  struct node{
7     int l, r, v;
8 } tree[maxn << 5];
9 void up(int now){
10     tree[now].v = tree[tree[now].l].v + tree[tree[now].r].v;
11 }
12 int build_tree(int l, int r){
13     int pos = ++tot;
14     if(l == r){
15         tree[pos].v = a[l];
16         return pos;
17     }
18     int mid = l + (r - l) / 2;
19     tree[pos].l = build_tree(l, mid);
20     tree[pos].r = build_tree(mid + 1, r);
21     up(pos);
22     return pos;
23 }
24 int update(int now, int l, int r, int tar, int v){
25     int pos = ++tot;
26     if(l == r){
27         tree[pos].v = v;
28         return pos;
29     }
30     int mid = l + (r - l) / 2;
31     tree[pos].l = tree[now].l;
32     tree[pos].r = tree[now].r;
33     up(pos);
34     if(tar <= mid) tree[pos].l = update(tree[now].l, l, mid, tar, v);
35     else tree[pos].r = update(tree[now].r, mid + 1, r, tar, v);
36     return pos;
37 }
38 int query(int now, int l, int r, int q){
39     if(l == r){
40         return now;
41     }
42     int mid = l + (r - l) / 2;
43     if(q <= mid) return query(tree[now].l, l, mid, q);
44     else return query(tree[now].r, mid + 1, r, q);

```

```

45 }
46 int main(){
47     int n,m;
48     while(~scanf("%d %d", &n, &m)){
49         tot=0;
50         for(int i=1;i<=n;i++){
51             scanf("%d", &a[i]);
52         }
53         root[0]=build_tree(1,n);
54         int u,v,l,k;
55         for(int i=1;i<=m;i++){
56             scanf("%d %d %d", &u, &k, &l);
57             if(k==1){
58                 scanf("%d", &v);
59                 root[i]=update(root[u],1,n,l,v);
60             }else{
61                 printf("%d\n", tree[query(root[u],1,n,l)].v);
62                 root[i]=root[u];
63             }
64         }
65     }
66     return 0;
67 }
68

```

主席树

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define maxn 200010
4  int n,m,tot;
5  int a[maxn],b[maxn],root[maxn];
6  struct node{
7     int l,r,v;
8 }tree[maxn<<5];
9 int build_tree(int l,int r){
10     int pos=++tot;
11     tree[pos].v=0;
12     if(l==r){
13         return pos;
14     }
15     int mid=l+(r-1)/2;
16     tree[pos].l=build_tree(l,mid);
17     tree[pos].r=build_tree(mid+1,r);
18     return pos;
19 }
20 int update(int now,int tar,int l,int r){
21     int pos=++tot;
22     if(l==r){
23         tree[pos].v=tree[now].v+1;
24         return pos;
25     }
26     int mid=l+(r-1)/2;
27     tree[pos].l=tree[now].l;
28     tree[pos].r=tree[now].r;
29     if(tar<=mid) tree[pos].l=update(tree[now].l,tar,l,mid);
30     else tree[pos].r=update(tree[now].r,tar,mid+1,r);
31     tree[pos].v=tree[tree[pos].l].v+tree[tree[pos].r].v;
32     return pos;
33 }
34 int query(int now,int last,int k,int l,int r){
35     if(l==r){
36         return l;
37     }
38     int mid=l+(r-1)/2;
39     int cnt=tree[tree[now].l].v-tree[tree[last].l].v;
40     if(k<=cnt) return query(tree[now].l,tree[last].l,k,l,mid);
41     else return query(tree[now].r,tree[last].r,k-cnt,mid+1,r);
42 }
43 int main(){

```



```

44     int n,m;
45     while(~scanf("%d %d", &n, &m)){
46         tot=0;
47         for(int i=1;i<=n;i++){
48             scanf("%d", &a[i]);
49             b[i]=a[i];
50         }
51         sort(b+1,b+1+n);
52         int res=unique(b+1,b+1+n)-(b+1);
53         for(int i=1;i<=n;i++){
54             a[i]=lower_bound(b+1,b+1+res,a[i])-b;
55         }
56         root[0]=build_tree(1,res);
57         for(int i=1;i<=n;i++){
58             root[i]=update(root[i-1],a[i],1,res);
59         }
60         int l,r,k;
61         while(m--){
62             scanf("%d %d %d", &l, &r, &k);
63             printf("%d\n", b[query(root[r],root[l-1],k,1,res)]);
64         }
65     }
66     return 0;
67 }
68

```

树状数组

```

1  #include <iostream>
2  #include <algorithm>
3  #include <cstring>
4  #include <cstdio>
5  using namespace std;
6  #define maxn 500010
7  int n;
8  int a[maxn];
9  int t[maxn*4];
10 struct node{
11     int x;
12     int id;
13 }f[maxn];
14 bool cmp(node a, node b){
15     return a.x<b.x;
16 }
17 void update(int x,int val){
18     while(x<=n){
19         t[x]+=val;
20         x+=x&(-x);
21     }
22 }
23 int sum(int x){
24     int ans=0;
25     while(x>=1){
26         ans+=t[x];
27         x-=x&(-x);
28     }
29     return ans;
30 }
31 int main(){
32     while(~scanf("%d", &n),n){
33         for(int i=1;i<=n;i++){
34             scanf("%d", &f[i].x);
35             f[i].id=i;
36         }
37         sort(f+1,f+n+1,cmp);
38         for(int i=1;i<=n;i++) a[i]=f[i].id;
39         memset(t,0,sizeof(t));
40         long long ans=0;
41         for(int i=1;i<=n;i++){
42             update(a[i],1);

```

```
43     ans+=(i-sum(a[i]));
44     }
45     printf("%lld\n", ans);
46 }
47 return 0;
48 }
```

树状数组维护差分数组

```
1  /*
2  【模板】树状数组 2
3  P3368
4  https://www.luogu.org/problem/P3368
5  解法： 树状数组维护差分数组
6  */
7  #include <bits/stdc++.h>
8  using namespace std;
9  #define ll long long
10 #define maxn 1000000
11 ll a[maxn];
12 ll tree[maxn<<1];
13 ll n,m;
14 inline long long read()//快读
15 {
16     long long x=0,f=1;char c=getchar();
17     while(!isdigit(c)){if(c=='-')f=-1;c=getchar();}
18     while(isdigit(c)){x=x*10+c-48;c=getchar();}
19     return x*f;
20 }
21 void update(int x,int val){
22     while(x<=n){
23         tree[x]+=val;
24         x+=x&(-x);
25     }
26 }
27 long long sum(long long x){
28     long long ans=0;
29     while(x>=1){
30         ans+=tree[x];
31         x-=x&(-x);
32     }
33     return ans;
34 }
35 int main(){
36     int s,l,r,x;
37     n=read();
38     m=read();
39     memset(tree,0,(n+1)*sizeof (ll));
40     for(int i=1;i<=n;i++){
41         a[i]=read();
42         update(i,a[i]-a[i-1]);
43     }
44     while(m--){
45         s=read();
46         if(s==2){
47             x=read();
48             printf("%lld\n", sum(x));
49         }else{
50             l=read();
51             r=read();
52             x=read();
53             update(l,x);
54             update(r+1,-x);
55         }
56     }
57 }
```

扫描线

```

1  /*
2  OverLapping Rectangles
3  2017 ACM-ICPC 亚洲区（南宁赛区）网络赛
4  https://nanti.jisuanke.com/t/A1282
5  题面：
6      n个矩形，给出左下角的坐标和右上角的坐标，算出n个矩形所覆盖的面积
7  输入：
8      2
9      0 0 2 2
10     1 1 3 3
11     3
12     0 0 1 1
13     2 2 3 3
14     4 4 5 5
15     0
16 输出：
17     7
18     3
19  */
20 #include <cstdio>
21 #include <cstring>
22 #include <algorithm>
23 #include <iostream>
24 using namespace std;
25 #define maxn 5010
26 struct node{
27     int ss;
28     double l,r,h;
29     node(double l1=0,double r1=0,double h1=0,int ss1=0){
30         l=l1;
31         r=r1;
32         h=h1;
33         ss=ss1;
34     }
35 }edge[maxn<<2];
36 int add[maxn];
37 double x[maxn<<2],sum[maxn<<2];
38 bool cmp(node a,node b){
39     return a.h<b.h;
40 }
41 void pushup(int now,int l,int r){
42     if(add[now]) sum[now]=x[r+1]-x[l];
43     else if(l==r) sum[now]=0;
44     else sum[now]=sum[now<<1]+sum[now<<1|1];
45 }
46 void update(int now,int L,int R,int val,int l,int r){
47     if(L<=l&&r<=R){
48         add[now]+=val;
49         pushup(now,l,r);
50         return ;
51     }
52     int mid=(l+r)>>1;
53     if(L<=mid) update(now<<1,L,R,val,l,mid);
54     if(R>mid) update(now<<1|1,L,R,val,mid+1,r);
55     pushup(now,l,r);
56 }
57 int main(){
58     int n;
59     double x1,x2,y1,y2,ans;
60     while(cin>>n && n){
61         ans=0;
62         int top=0,l,r;
63         for(int i=0;i<n;i++){
64             cin>>x1>>y1>>x2>>y2;
65             x[top]=x1;
66             edge[top++]=node(x1,x2,y1,1);
67             x[top]=x2;
68             edge[top++]=node(x1,x2,y2,-1);
69         }
70         sort(x,x+top);

```

```

71     sort(edge,edge+top,cmp);
72     int k = 1;
73     for(int i=1; i<top; i++)
74     {
75         if(x[i] != x[i-1])
76             x[k++] = x[i];
77     }
78     memset(add,0,sizeof(add));
79     memset(sum,0,sizeof(sum));
80     for(int i=0; i<top-1; i++){
81         l=lower_bound(x,x+k,edge[i].l)-x;
82         r=lower_bound(x,x+k,edge[i].r)-x-1;
83         update(1,l,r,edge[i].ss,0,k-1);
84         ans+=(sum[1]*(edge[i+1].h-edge[i].h));
85     }
86     cout<<(int)ans<<endl;
87 }
88 cout<<"*"<<endl;
89 return 0;
90 }
91
92 /*
93 The beautiful values of the palace
94 The Preliminary Contest for ICPC Asia Nanjing 2019 A.
95 https://nanti.jisuanke.com/t/41298
96 题意: 有一个n*n (n为奇数)的矩阵, 沿着方阵中心螺旋下降, 有m个地方是要建宫殿的,
97 宫殿的价值是土地价值的数字总和 例如: 123213 (1+2+3+2+1+3=12)
98 p次询问 问一个区间的所以宫殿价值总和
99 样例输入:
100 1
101 3 4 4
102 1 1
103 2 2
104 3 3
105 2 3
106 1 1 1 1
107 2 2 3 3
108 1 1 3 3
109 1 2 2 3
110 样例输出:
111 5
112 18
113 23
114 17
115 解法: 因为n比较大, 所以只能用数学方法计算出要建宫殿的地方的价值
116 然后因为p比较大, 所以要用到扫描线来处理
117 */
118 #include <bits/stdc++.h>
119 using namespace std;
120 #define maxn 1000100
121 #define ll long long
122 int n,m,p,cnt;
123 int tree[maxn];
124 int sum[maxn],res[maxn];
125 struct node{
126     int f,L,R,h,id;
127     int val;
128 }edge[maxn];
129 int get(int x, int y){
130     long long centre = (long long)n*n;
131     long long cen = n/2+1;
132     long long val;
133     if (x==cen && y==cen){
134         int ans=0;
135         val=centre;
136         while(val){
137             ans+=val%10;
138             val=val/10;
139         }
140         return ans;
141     }

```

```

142     long long Max = max(abs(x-cen), abs(y-cen));
143     val = centre - Max * (4*Max +1);
144     if((y == (cen-Max) && x >= cen) || x == cen + Max) val -= (x-cen) + (y-cen+Max);
145     else {
146         if(y != (cen-Max)) val += (x-cen + Max) + (y-cen + Max) + Max;
147         else val += cen - x;
148     }
149     int ans=0;
150     while(val){
151         ans+=val%10;
152         val=val/10;
153     }
154     return ans;
155 }
156 bool cmp(node a,node b){
157     if(a.h==b.h){
158         if((a.f==1&&b.f==0)|| (b.f==1&&a.f==0)) return a.f>b.f;
159         else return a.f<b.f;
160     }
161     return a.h<b.h;
162 }
163 void update(int pow,int val){
164     while(pow<maxn){
165         tree[pow]+=val;
166         pow+=(-pow)&pow;
167     }
168 }
169 int query(int pow){
170     int ans=0;
171     while(pow>0){
172         ans+=tree[pow];
173         pow-=(-pow)&pow;
174     }
175     return ans;
176 }
177 int main(){
178     int T;
179     int x1,x2,y1,y2,x,y;
180     cin>>T;
181     while(T--){
182         scanf("%d %d %d", &n, &m, &p);
183         cnt=0;
184         memset(tree,0,sizeof tree);
185         while(m--){
186             scanf("%d %d", &x, &y);
187             edge[cnt++]=(node){0,x,0,y,0,get(x,y)};
188         }
189         for(int i=1;i<=p;i++){
190             scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
191             edge[cnt++]=(node){1,x1,x2,y1,i,0};
192             edge[cnt++]=(node){2,x1,x2,y2,i,0};
193         }
194         sort(edge,edge+cnt,cmp);
195         for(int i=0;i<cnt;i++){
196             if(edge[i].f==0) update(edge[i].L,edge[i].val);
197             else if(edge[i].f==1) sum[edge[i].id]=query(edge[i].R)-query(edge[i].L-1);
198             else res[edge[i].id]=query(edge[i].R)-query(edge[i].L-1)-sum[edge[i].id];
199         }
200         for(int i=1;i<=p;i++){
201             printf("%d\n", res[i]);
202         }
203     }
204     return 0;
205 }

```

01字典树

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 #define maxn 100000

```

```

4  #define ll long long
5  int tol;
6  ll val[32*maxn];
7  int ch[32*maxn][2];
8  void init(){
9      tol=1;
10     ch[0][0]=ch[0][1]=0;
11 }
12 void insert(ll x){
13     int u=0;
14     for(int i=32;i>=0;i--){
15         int v=(x>>i)&1;
16         if(!ch[u][v]){
17             ch[tol][0]=ch[tol][1]=0;
18             val[tol]=0;
19             ch[u][v]=tol++;
20         }
21         u=ch[u][v];
22     }
23     val[u]=x;
24 }
25 ll query(ll x){
26     int u=0;
27     for(int i=32;i>=0;i--){
28         int v=(x>>i)&1;
29         if(ch[u][v^1]) u=ch[u][v^1];
30         else u=ch[u][v];
31     }
32     return val[u];
33 }
34 int main(){
35     int T,n,m;
36     ll a;
37     scanf("%d",&T);
38     for(int s=1;s<=T;s++){
39         init();
40         scanf("%d %d", &n, &m);
41         for(int i=0;i<n;i++){
42             scanf("%lld", &a);
43             insert(a);
44         }
45         printf("Case #%d:\n", s);
46         for(int i=0;i<m;i++){
47             scanf("%lld", &a);
48             printf("%d\n", query(a));
49         }
50     }
51 }
52
53 /*
54 Chip Factory
55 ACM/ICPC 2015 Changchun (Trie)
56 https://nanti.jisuanke.com/t/A1941
57 题面:
58 有一个数组, 让你选取三个数, 使其中两个数相加异或上另外一个数值最大。
59 很明显的异或最大要用01字典树, 问题是要选取三个数。我们可以提前把所有的数字插入到01字典树上,
60 每次枚举其中两个数时, 先在字典树上面删除这两个数, 然后再查询。问题又来了, 我们怎么删除了, 因为我们用的是公共前缀,
61 所以我们可以对01字典树上的边进行计数, 每次插入 +1, 删除 -1, 如此就可以完成删除操作了。
62 样例输入:
63     2
64     3
65     1 2 3
66     3
67     100 200 300
68 样例输出:
69     6
70     400
71 */
72 #include <bits/stdc++.h>
73 using namespace std;
74 #define maxn 1000

```

```

75 #define ll long long
76 int tol;
77 ll val[30*maxn];
78 int ch[30*maxn][2];
79 int nn[30*maxn][2];
80 void init(){
81     tol=1;
82     memset(nn,0,sizeof nn);
83     ch[0][0]=ch[0][1]=0;
84 }
85 void insert(ll x){
86     int u=0;
87     for(int i=30;i>=0;i--){
88         int v=(x>>i)&1;
89         nn[u][v]++;
90         if(!ch[u][v]){
91             ch[tol][0]=ch[tol][1]=0;
92             val[tol]=0;
93             ch[u][v]=tol++;
94         }
95         u=ch[u][v];
96     }
97     val[u]=x;
98 }
99 ll query(ll x){
100     int u=0;
101     for(int i=30;i>=0;i--){
102         int v=(x>>i)&1;
103         if(ch[u][v^1]&&nn[u][v^1]) u=ch[u][v^1];
104         else u=ch[u][v];
105     }
106     return val[u];
107 }
108 void pp(){
109     for(int i=0;i<tol;i++){
110         cout<<i<<" " <<nn[i][0]<<" " <<nn[i][1]<<endl;
111     }
112 }
113 ll solve(ll x,ll y){
114     ll xx=x,yy=y;
115     ll num=x+y;
116     ll u=0;
117     for(int i=30;i>=0;i--){
118         int v=(x>>i)&1;
119         nn[u][v]--;
120         u=ch[u][v];
121     }
122     u=0;
123     for(int i=30;i>=0;i--){
124         int v=(y>>i)&1;
125         nn[u][v]--;
126         u=ch[u][v];
127     }
128     ll ans=num^query(num);
129     u=0;
130     for(int i=30;i>=0;i--){
131         int v=(xx>>i)&1;
132         nn[u][v]++;
133         u=ch[u][v];
134     }
135     u=0;
136     for(int i=30;i>=0;i--){
137         int v=(yy>>i)&1;
138         nn[u][v]++;
139         u=ch[u][v];
140     }
141     return ans;
142 }
143 ll a[maxn];
144 int main(){
145     int T,n,m;

```

```

146     scanf("%d",&T);
147     while(T--){
148         init();
149         scanf("%d", &n);
150         for(int i=0;i<n;i++){
151             scanf("%lld", &a[i]);
152             insert(a[i]);
153         }
154         int ans=0,num;
155         for(int i=0;i<n;i++){
156             for(int j=i+1;j<n;j++){
157                 num=solve(a[i],a[j]);
158                 if(ans<num) ans=num;
159             }
160         }
161         cout<<ans<<endl;
162     }
163 }

```

字典树

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  struct node {
4      int cnt;
5      node* child[26];
6  };
7  node* root;
8  void init()
9  {
10     root=new node();
11 }
12 void insert(const char s[])
13 {
14     node* now=root;
15     for(int i=0;s[i];i++){
16         int go=s[i]-'a';
17         if(now->child[go]==NULL)
18             now->child[go]=new node();
19         now=now->child[go];
20         now->cnt++;
21     }
22 }
23 int query(const char s[])
24 {
25     node* now=root;
26     for(int i=0;s[i];i++){
27         int go=s[i]-'a';
28         if(now->child[go]==NULL) return -1;
29         now=now->child[go];
30     }
31     return now->cnt;
32 }
33 void del(const char s[])
34 {
35     node* now=root;
36     for(int i=0;s[i];i++){
37         int go=s[i]-'a';
38         if(now->child[go]->cnt==1){
39             now->child[go]=NULL;
40             return;
41         }
42         now=now->child[go];
43         now->cnt--;
44     }
45 }
46 int main()
47 {
48     init();
49     insert("abcdef");

```



```

50     insert("abccba");
51     cout << query("abc") << endl;
52     cout << query("abcc") << endl;
53     cout << query("abs") << endl;
54     cout << query("abcdef") << endl;
55     del("abcdef");
56     cout << query("abcdef") << endl;
57     cout<<query("abc")<<endl;
58     return 0;
59 }

```

并查集

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define maxn 100000
4  int n,m;
5  int u,v,k;
6  int fa[maxn];
7  void init(){
8      for(int i=0;i<=n;i++)
9          fa[i]=i;
10 }
11 int root(int x){
12     return fa[x]==x?x:fa[x]=root(fa[x]);
13 }
14 void unite(int u,int v){
15     u=root(u);
16     v=root(v);
17     if(u!=v)
18         fa[v]=u;
19 }
20 bool ailke(int u,int v){
21     return root(u)==root(v);
22 }
23 int main(){
24     cin>>n>>m;
25     init();
26     for(int i=0;i<m;i++){
27         cin>>k>>u>>v;
28         if(k==1) unite(u,v);
29         if(k==2){
30             if(ailke(u,v)){
31                 cout<<"Y"<<endl;
32             }else{
33                 cout<<"N"<<endl;
34             }
35         }
36     }
37     return 0;
38 }
39

```

带权并查集

```

1  /*
2  题面:
3      动物王国中有三类动物A,B,C，这三类动物的食物链构成了有趣的环形。A吃B， B吃C， C吃A。
4      现有N个动物，以1—N编号。每个动物都是A,B,C中的一种，但是我们并不知道它到底是哪一种。
5      有人用两种说法对这N个动物所构成的食物链关系进行描述:
6      第一种说法是"1 X Y"，表示X和Y是同类。
7      第二种说法是"2 X Y"，表示X吃Y。
8      此人对N个动物，用上述两种说法，一句接一句地说出K句话，这K句话有的是真的，有的是假的。
9      当一句话满足下列三条之一时，这句话就是假话，否则就是真话。
10     1) 当前的话与前面的某些真的话冲突，就是假话；
11     2) 当前的话中X或Y比N大，就是假话；
12     3) 当前的话表示X吃X，就是假话。
13     你的任务是根据给定的N (1 ≤ N ≤ 50,000) 和K句话 (0 ≤ K ≤ 100,000)，输出假话的总数。
14 Input
15     第一行是两个整数N和K，以一个空格分隔。

```

16 以下K行每行是三个正整数 D, X, Y, 两数之间用一个空格隔开, 其中D表示说法的种类。

17 若D=1, 则表示X和Y是同类。

18 若D=2, 则表示X吃Y。

19 Output

20 只有一个整数, 表示假话的数目。

21 Sample Input

22 100 7

23 1 101 1

24 2 1 2

25 2 2 3

26 2 3 3

27 1 1 3

28 2 3 1

29 1 5 5

30 Sample Output

31 3

32 */

33 //group[0] 同类

34 //group[1] 被根吃

35 //group[2] 吃根

36 #include<iostream>

37 #include<cstdio>

38 #include<cstring>

39 using namespace std;

40 #define maxn 1000000

41 int fa[maxn];

42 int group[maxn];

43 int n,m;

44 int ans;

45 int d,x,y;

46 void init(){

47 for(int i=0;i<=n;i++){

48 fa[i]=i;

49 group[i]=0;

50 }

51 ans=0;

52 }

53 int root(int x){

54 if(fa[x]==x)return x;

55 int r=root(fa[x]);

56 group[x]=(group[x]+group[fa[x]])%3;

57 return fa[x]=r;

58 }

59 int main(){

60 scanf("%d%d",&n,&m);

61 init();

62 while(m--){

63 scanf("%d%d%d",&d,&x,&y);

64 if(x>n||y>n||(d==2&&x==y)){

65 ans++;

66 continue;

67 }

68 int fx=root(x);

69 int fy=root(y);

70 if(fx==fy){

71 if(d==1) if(group[x]!=group[y]) ans++;

72 if(d==2) if((3+group[y]-group[x])%3!=1) ans++;

73 }else{

74 group[fy]=((d-1)+3+group[x]-group[y])%3;

75 fa[fy]=fx;

76 }

77 }

78 cout<<ans<<endl;

79 return 0;

80 }

81

有 0 个人打赏

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客

编程语言大PK，你选谁？

关闭