

原创 SUST暑期集训题解（可持久化数据结构）

2019-08-24 15:43:07 laaahu_ 阅读数 38 更多

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/laaahu/article/details/100053414>

A 可持久化线段树

```
1  #include<iostream>
2  #include<cstdio>
3  #include<cstring>
4  #include<algorithm>
5  #include<cmath>
6  using namespace std;
7  const int maxn=1e6+10;
8  const int M=maxn*40;
9  int a[maxn];
10 int T[maxn],lson[M],rson[M],c[M];
11 int n,m,tot;
12
13 int build(int l,int r){
14     int root = tot++;
15     c[root]=0;
16     if(l!=r){
17         int mid=(l+r)>>1;
18         lson[root]=build(l,mid);
19         rson[root]=build(mid+1,r);
20     }
21     if(l==r)
22         c[root]=a[l];
23     return root;
24 }
25
26 int update(int root,int pos,int val){
27     int newroot=tot++,tmp=newroot;
28     c[newroot]=0;
29     int l=1,r=n;
30     while(l<r){
31         int mid = (l+r)>>1;
32         if(pos<=mid){
33             lson[newroot]=tot++;rson[newroot]=rson[root];
34             newroot=lson[newroot];root=lson[root];
35             r=mid;
36         }
37         else{
38             rson[newroot]=tot++;lson[newroot]=lson[root];
39             newroot=rson[newroot];root=rson[root];
40             l=mid+1;
41         }
42         c[newroot]=0;
43     }
44     c[newroot]=val;
45     return tmp;
46 }
47
48 int query(int root,int pos){
49     int l=1,r=n;
50     while(l<r){
51         int mid=(l+r)>>1;
52         if(pos<=mid){
53             r=mid;
54             root=lson[root];
55         }
56         else{
57             l=mid+1;
58             root=rson[root];
59         }
60     }
```

猿衣酷

专属于程序员的卫衣

关闭

```

60     if(pos==1)
61         return c[root];
62 }
63 int main()
64 {
65     int cnt=0;
66     tot=0;
67     scanf("%d%d",&n,&m);
68     for(int i=1;i<=n;i++)
69         scanf("%d",&a[i]);
70     T[cnt++]=build(1,n);
71     for(int i=0;i<m;i++){
72         int aa,bb,cc,dd;
73         scanf("%d%d",&aa,&bb);
74         if(bb==1){
75             scanf("%d%d",&cc,&dd);
76             T[cnt++]=update(T[aa],cc,dd);
77         }
78         else{
79             scanf("%d",&cc);
80             printf("%d\n",query(T[aa],cc));
81             T[cnt++]=T[aa];
82         }
83     }
84 }
85

```

B 区间第K小

```

1  #include<iostream>
2  #include<cstdio>
3  #include<cstring>
4  #include<algorithm>
5  #include<cmath>
6  using namespace std;
7  const int maxn = 2e5+10;
8  const int M = maxn*40;
9  int n,m,q,tot;
10 int a[maxn],t[maxn];
11 int T[maxn],lson[M],rson[M],c[M];
12 void init_hash(){
13     for(int i = 1;i <= n;i++){
14         t[i]=a[i];
15     }
16     sort(t+1,t+n+1);
17     m=unique(t+1,t+n+1)-t-1;
18 }
19 int Hash(int x){
20     return lower_bound(t+1,t+1+m,x)-t;
21 }
22 int build(int l,int r){
23     int root=tot++;
24     c[root]=0;
25     if(l==r){
26         int mid = (l+r)>>1;
27         lson[root]=build(l,mid);
28         rson[root]=build(mid+1,r);
29     }
30     return root;
31 }
32 int update(int root,int pos,int val){
33     int newroot=tot++,tmp=newroot;
34     c[newroot]=c[root]+val;
35     int l=1, r=m;
36     while(l<r){
37         int mid=(l+r)>>1;
38         if(pos<=mid){
39             lson[newroot] = tot++;rson[newroot] = rson[root];
40             newroot = lson[newroot]; root = lson[root];
41             r=mid;

```

猿衣酷

专属于程序员的卫衣

关闭

```

42     }
43     else{
44         rson[newroot] = tot++;lson[newroot] = lson[root];
45         newroot = rson[newroot]; root = rson[root];
46         l=mid+1;
47     }
48     c[newroot]=c[root]+val;
49 }
50 return tmp;
51 }
52
53 int query(int left_root,int right_root,int k){
54     int l = 1,r = m;
55     while(l<r){
56         int mid = (l+r)>>1;
57         if(c[lson[left_root]]-c[lson[right_root]]>=k){
58             r=mid;
59             left_root=lson[left_root];
60             right_root=lson[right_root];
61         }
62         else{
63             l=mid+1;
64             k-=c[lson[left_root]]-c[lson[right_root]];
65             left_root = rson[left_root];
66             right_root = rson[right_root];
67         }
68     }
69     return l;
70 }
71 int main()
72 {
73     while(~scanf("%d%d",&n,&q)){
74         tot=0;
75         for(int i=1;i<=n;i++){
76             scanf("%d",&a[i]);
77         }
78         init_hash();
79         T[n+1]=build(1,m);
80         for(int i=n;i--){
81             int pos = Hash(a[i]);
82             T[i]=update(T[i+1],pos,1);
83         }
84         while(q--){
85             int l,r,k;
86             scanf("%d%d%d",&l,&r,&k);
87             printf("%d\n",t[query(T[l],T[r+1],k)]);
88         }
89     }
90     return 0;
91 }
92

```

C 可持久化并查集

```

1  #include<iostream>
2  #include<cstdio>
3  #include<algorithm>
4  #include<cstring>
5  #include<cmath>
6  using namespace std;
7  const int maxn = 2e3+10;
8  const int M=maxn*80;
9  int T[maxn],lson[M],rson[M],per[M],dep[M];
10 int n,m,tot;
11
12 void build(int &rt,int l,int r)
13 {
14     rt=++tot;
15     if(l==r){
16         per[rt]=l;

```

猿衣酷

专属于程序员的卫衣

关闭

```

17     return;
18 }
19 int mid=(l+r)>>1;
20 build(lson[rt],l,mid);
21 build(rson[rt],mid+1,r);
22 }
23 void merge(int last,int &rt,int l,int r,int pos,int fa)
24 {
25     rt=++tot;lson[rt]=lson[last],rson[rt]=rson[last];
26     if(l==r){
27         per[rt]=fa;
28         dep[rt]=dep[last];
29         return ;
30     }
31     int mid=(l+r)>>1;
32     if(pos<=mid) merge(lson[last],lson[rt],l,mid,pos,fa);
33     else merge(rson[last],rson[rt],mid+1,r,pos,fa);
34 }
35 void update(int rt,int l,int r,int pos)
36 {
37     if(l==r){
38         dep[rt]++;
39         return;
40     }
41     int mid=(l+r)>>1;
42     if(pos<=mid)update(lson[rt],l,mid,pos);
43     else update(rson[rt],mid+1,r,pos);
44 }
45 int query(int rt,int l,int r,int pos)
46 {
47     if(l==r)return rt;
48     int mid=(l+r)>>1;
49     if(pos<=mid) return query(lson[rt],l,mid,pos);
50     else return query(rson[rt],mid+1,r,pos);
51 }
52 int find(int rt,int pos)
53 {
54     int now=query(rt,1,n,pos);
55     if(per[now]==pos)return now;
56     return find(rt,per[now]);
57 }
58 int main()
59 {
60     tot=0;
61     scanf("%d",&n,&m);
62     build(T[0],1,n);
63     for(int i=1;i<=m;i++){
64         int opt,x,y;
65         scanf("%d",&opt,&x);
66         if(opt==1){
67             scanf("%d",&y);
68             T[i]=T[i-1];
69             int posx=find(T[i],x);
70             int posy=find(T[i],y);
71             if(per[posx]!=per[posy]){
72                 if(dep[posx]>dep[posy])swap(posx,posy);
73                 merge(T[i-1],T[i],1,n,per[posx],per[posy]);
74                 if(dep[posx]==dep[posy])update(T[i],1,n,per[posy]);
75             }
76         }
77         else if(opt==2){
78             T[i]=T[x];
79         }
80         else if(opt==3){
81             scanf("%d",&y);
82             T[i]=T[i-1];
83             int posx,posy;
84             posx=find(T[i],x);
85             posy=find(T[i],y);
86             if(per[posx]==per[posy])printf("1\n");
87             else printf("0\n");

```

猿衣酷

专属于程序员的卫衣

关闭

```
88 |     }  
89 | }  
90 |     return 0;  
91 | }
```

有 0 个人打赏 文章最后发布于: 2019-08-24 15:43:07

猿衣酷

专属于程序员的卫衣

关闭