

Nexus_Raphael的模板

字符串部分

KMP

string从1开始数

```
1 int L=s.length();
2 int j=0;
3 for(int i=2;i<L;i++){
4     while(j&& s[j+1]!=s[i]) j=nxt[j];
5     if(s[j+1]==s[i]) j++;
6     nxt[i]=j;
7 }
```

Manacher(马拉车)

```
1 int mnc_mxpali(string s){
2     string s1="$";
3     for(int i=0;i<s.length();i++){
4         s1=s1+'#';
5         s1=s1+s[i];
6     }
7     s1=s1+"#*";
8     vector<int>p(s1.length());
9     int c=0,r=0;
10    for(int i=1;i<s1.length();i++){
11        if(i>r) p[i]=0;
12        else p[i]=min(r-i,p[2*c-i]);
13        while(s1[i-p[i]-1]==s1[i+p[i]+1]) p[i]++;
14        if(p[i]+i>r){
15            r=p[i]+i;
16            c=i;
17        }
18    }
19    int mx=-1;
20    for(auto x:p){
21        mx=max(x,mx);
22    }
23    return mx;
24 }
25 /*****/
26 int d1[N],d2[N];
27
28 void mnc_odd(string s){
29     int len=0;
30     int n=s.length();
31     for(int i=0,l=0,r=-1;i<n;i++){
32         if(i>r) len=1; //越过边界就置为1
33         else len=min(d1[l+r-i],r-i+1); //否则置为对称处的长，不越界
34         while(i-len>=0&&i+len<n&&s[i-len]==s[i+len]) len++;
35     }
```

```

35         d1[i]=len;
36         len--;
37         if(i+len>r){
38             l=i-len;
39             r=i+len;
40         }
41     }
42 }
43
44 void mnc_even(string s){//切记这个i是偏右的那个位置
45                             //也即i左边0.5格才是所算回文串的中心
46     int len=0;
47     int n=s.length();
48     for(int i=0,l=0,r=-1;i<n;i++){
49         if(i>r) len=0;
50         else len=min(d2[l+r-i+1],r-i+1);
51         while(i-len-1>=0&&i+len<n&&s[i-len-1]==s[i+len]) len++;
52         d2[i]=len;
53         len--;
54         if(i+len>r){
55             l=i-len-1;
56             r=i+len;
57         }
58     }
59 }

```

数据结构部分

树状数组

树状数组都是从1开始存,下面是求逆序数的代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define int long long
4  #define endl '\n'
5  const int N=5e5+5;
6  struct fac{
7      int val,pos;
8  }a[N];
9  int b[N];
10 int tree[N];
11 int n;
12
13 bool cmp(fac x,fac y){
14     if(x.val==y.val) return x.pos<y.pos;
15     return x.val<y.val;
16 }
17
18 int lowbit(int x){return x&-x;}
19
20 void ins(int p,int x){
21     while(p<=n){
22         tree[p]+=x;
23         p+=lowbit(p);
24     }

```

```

25 }
26
27 int qry(int p){
28     int ret=0;
29     while(p>=1){
30         ret+=tree[p];
31         p-=lowbit(p);
32     }
33     return ret;
34 }
35
36 void solve(){
37     cin>>n;
38     for(int i=1;i<=n;i++){
39         cin>>a[i].val;
40         a[i].pos=i;
41     }
42     sort(a+1,a+n+1,cmp);
43     for(int i=1;i<=n;i++){
44         b[a[i].pos]=i;
45     }
46     int ans=0;
47     for(int i=1;i<=n;i++){
48         ins(b[i],1);
49         ans+=i-qry(b[i]);
50     }
51     cout<<ans<<endl;
52 }
53
54 signed main(){
55     ios::sync_with_stdio(false);
56     cin.tie(0);
57     int T=1;
58     // cin>>T;
59     while(T--){
60         solve();
61         return 0;
62     }

```

线段树

下面分别是区间求和+区间加数 和 区间求和和区间加/乘数的代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long ll;
4  #define MOD 1000000007
5  const int MAX=5e5+5;
6  ll tree[MAX<<2],g[MAX<<2];//g[] 是懒标记
7  ll a[MAX];
8  ll n,m;
9
10 void build(ll o,ll l,ll r){
11     if(l==r){tree[o]=a[l];return ;}
12     ll mid=(l+r)>>1;
13     build(o<<1,l,mid);
14     build(o<<1|1,mid+1,r);

```

```

15     tree[o]=tree[o<<1]+tree[o<<1|1];
16 }
17
18 void pushdown(ll o,ll l,ll r){
19     if(g[o]==0) return ;
20     ll mid=(l+r)>>1;
21     g[o<<1]+=g[o];
22     g[o<<1|1]+=g[o];
23     tree[o<<1]+=g[o]*(mid-l+1);
24     tree[o<<1|1]+=g[o]*(r-mid);
25     g[o]=0;
26 }
27
28 void add(ll o,ll l,ll r,ll x,ll y,ll k){
29     if(l>y||r<x) return;
30     if(l>=x&&r<=y){
31         tree[o]+=k*(r-l+1);
32         g[o]+=k;
33         return ;
34     }
35     ll mid=(l+r)>>1;
36     pushdown(o,l,r);
37     if(x<=mid) add(o<<1,l,mid,x,y,k);
38     if(y>=mid+1) add(o<<1|1,mid+1,r,x,y,k);
39     tree[o]=tree[o<<1]+tree[o<<1|1];
40 }
41
42 ll sum(ll o,ll l,ll r,ll L,ll R){
43     ll ans=0;
44     if(l>R||r<L) return 0;
45     if(l>=L&&r<=R) return tree[o];
46     pushdown(o,l,r);
47     ll mid=(l+r)>>1;
48     if(L<=mid) ans+=sum(o<<1,l,mid,L,R);
49     if(R>=mid+1) ans+=sum(o<<1|1,mid+1,r,L,R);
50     return ans;
51 }
52
53 void solve(){
54     cin>>n>>m;
55     for(int i=1;i<=n;i++) cin>>a[i];
56     build(1,1,n);
57
58     for(int i=0;i<m;i++){
59         int opt;
60         cin>>opt;
61         if(opt==1){
62             ll x,y,k;
63             cin>>x>>y>>k;
64             add(1,1,n,x,y,k);
65         }else{
66             ll ans=0;
67             ll L,R;
68             cin>>L>>R;
69             ans+=sum(1,1,n,L,R);
70             cout<<ans<<endl;
71         }
72     }

```

```

73 }
74
75 int main(){
76     ios::sync_with_stdio(false);
77     cin.tie(0);
78     int T=1;
79     // cin>>T;
80     while(T--){
81         solve();
82     }
83     return 0;
84 }

```

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long ll;
4  const int N=1e5+5;
5  ll t[N<<2],a[N],ad[N<<2],mu[N<<2];
6  int n,q,m;
7
8  void build(ll o,ll l,ll r){
9      mu[o]=1;
10     if(l==r){t[o]=a[l]%m;return ;}
11     int mid=(l+r)>>1;
12     build(o<<1,l,mid);
13     build(o<<1|1,mid+1,r);
14     t[o]=(t[o<<1]+t[o<<1|1])%m;
15 }
16
17 void pushdown(ll o,ll l,ll r){
18     int mid=(l+r)>>1;
19     t[o<<1]=(t[o<<1]*mu[o]+ad[o]*(mid-l+1))%m;
20     t[o<<1|1]=(t[o<<1|1]*mu[o]+ad[o]*(r-mid))%m;
21
22     ad[o<<1]=(ad[o<<1]*mu[o]+ad[o])%m;
23     ad[o<<1|1]=(ad[o<<1|1]*mu[o]+ad[o])%m;
24
25     mu[o<<1]=(mu[o<<1]*mu[o])%m;
26     mu[o<<1|1]=(mu[o<<1|1]*mu[o])%m;
27
28     ad[o]=0;
29     mu[o]=1;
30 }
31
32 void add(ll o,int l,int r,int x,int y,int k){
33     if(x<=l&&y>=r){
34         t[o]=(t[o]+k*(r-l+1))%m;
35         ad[o]=(ad[o]+k)%m;
36         return ;
37     }
38     int mid=(l+r)>>1;
39     pushdown(o,l,r);
40     if(x<=mid) add(o<<1,l,mid,x,y,k);
41     if(y>mid) add(o<<1|1,mid+1,r,x,y,k);
42     t[o]=(t[o<<1]+t[o<<1|1])%m;
43 }
44

```

```

45 void mul(ll o,int l,int r,int x,int y,int k){
46     if(x<=l&&y>=r){
47         t[o]=t[o]*k%m;
48         mu[o]=mu[o]*k%m;
49         ad[o]=ad[o]*k%m;
50         return ;
51     }
52     pushdown(o,l,r);
53     int mid=(l+r)>>1;
54     if(x<=mid) mul(o<<1,l,mid,x,y,k);
55     if(y>mid) mul(o<<1|1,mid+1,r,x,y,k);
56     t[o]=(t[o<<1]+t[o<<1|1])%m;
57 }
58
59 ll qry(ll o,int l,int r,int L,int R){
60     if(L<=l&&r<=R) return t[o];
61     pushdown(o,l,r);
62     int mid=(l+r)>>1;
63     ll ans=0;
64     if(L<=mid) ans=(ans+qry(o<<1,l,mid,L,R))%m;
65     if(R>mid) ans=(ans+qry(o<<1|1,mid+1,r,L,R))%m;
66     return ans;
67 }
68
69 void solve(){
70     cin>>n>>q>>m;
71     for(int i=1;i<=n;i++) cin>>a[i];
72     build(1,1,n);
73     while(q--){
74         int opt;
75         cin>>opt;
76         if(opt==1){
77             int x,y,k;
78             cin>>x>>y>>k;
79             mul(1,1,n,x,y,k);
80         }else if(opt==2){
81             int x,y,k;
82             cin>>x>>y>>k;
83             add(1,1,n,x,y,k);
84         }else{
85             int L,R;
86             cin>>L>>R;
87             ll ans=0;
88             ans=qry(1,1,n,L,R);
89             cout<<ans%m<<endl;
90         }
91     }
92 }
93
94 signed main(){
95     ios::sync_with_stdio(false);
96     cin.tie(0);
97     int T=1;
98     // cin>>T;
99     while(T--){
100         solve();
101         return 0;
102     }

```

并查集

```
1 //这是信号站的代码,涉及到并查集
2 #include<bits/stdc++.h>
3 using namespace std;
4 #define int long long
5 #define ll long long
6 #define fi first
7 #define se second
8 #define vi vector<int>
9 #define vll vector<long long>
10 #define pii pair<int,int>
11 #define endl '\n'
12 #define MOD 1000000007
13 const int N=2e5+5;
14 int n,d;
15 int f[N],x[N],y[N],ok[N];
16
17
18 int getf(int v){
19     if(f[v]==v) return v;
20     f[v]=getf(f[v]);
21     return f[v];
22 }
23
24 void merge(int u,int v){
25     int fu=getf(u),fv=getf(v);
26     if(fu>fv) fu=fv;
27     else fv=fu;
28 }
29
30
31
32 void solve(){
33     cin>>n>>d;
34     memset(x,0,n+1);
35     memset(y,0,n+1);
36     memset(ok,0,n+1);
37     for(int i=1;i<=n;i++){
38         cin>>x[i]>>y[i];
39         f[i]=i;
40     }
41     char c;
42     while(cin>>c){
43         if(c=='o'){
44             int t;
45             cin>>t;
46             ok[t]=1;
47             for(int i=1;i<=n;i++){
48                 if(ok[i]&&(x[t]-x[i])*(x[t]-x[i])+(y[t]-y[i])*(y[t]-y[i])
49 <=d*d){
50                     merge(t,i);
51                 }
52             }
53         }
54     }
55 }
```

```

52         }else {
53             int a,b;
54             cin>>a>>b;
55             if(ok[a]&&ok[b]&&getf(a)==getf(b)) cout<<"SUCCESS\n";
56             else cout<<"FAIL\n";
57         }
58     }
59
60 }
61
62 signed main(){
63     ios::sync_with_stdio(false);
64     cin.tie(0);
65     int T=1;
66     // cin>>T;
67     while(T-->0)
68         solve();
69     return 0;
70 }

```

```

1  //这个是食物链的代码,算是并查集的深度使用
2  #include<bits/stdc++.h>
3  using namespace std;
4
5  const int N=2e5+5;
6  int n,r[N],p[N]; //r[]表示x和p[x]的关系
7                  //p[]表示x的父亲
8
9  int getf(int i){
10     int t;
11     if(p[i]==i) return i;
12     t=p[i];
13     p[i]=getf(p[i]);
14     r[i]=(r[t]+r[i])%3;
15     return p[i];
16 }
17
18 void merge(int x,int y,int h){
19     int a=getf(x),b=getf(y);
20     p[a]=b;
21     r[a]=(h+3+r[y]-r[x])%3;
22 }
23
24 void solve(){
25     int n,k;
26     cin>>n>>k;
27     int ans=0;
28     for(int i=1;i<=n;i++) p[i]=i;
29     while(k-->0){
30         int d,x,y;
31         cin>>d>>x>>y;
32         if(x>n||y>n){
33             ans++;
34             continue;
35         }
36         if(d==1){
37             if(getf(x)==getf(y)){

```



```

38         if(r[x]!=r[y]) ans++;
39     }
40     else merge(x,y,0);
41 }else{
42     if(getf(x)==getf(y)){
43         if(r[x]!=(r[y]+1)%3) ans++;
44     }
45     else merge(x,y,1);
46 }
47 }
48 cout<<ans<<endl;
49 }
50
51 signed main(){
52     ios::sync_with_stdio(false);
53     cin.tie(0);
54     int T=1;
55     // cin>>T;
56     while(T--){
57         solve();
58     }
59     return 0;
60 }

```

图论

tarjan

```

1 //HDU-1269迷宫城堡
2 #include<iostream>
3 #include<cstdlib>
4 #include<cstdio>
5 #include<algorithm>
6
7 using namespace std;
8 const int MAX=1e5;
9 struct node{
10     int to;
11     int next;
12 }edge[MAX+5];
13 int n,m;
14 int head[MAX+5],ans;
15 void addnode(int u,int v)
16 {
17     edge[ans].to=v;
18     edge[ans].next=head[u];
19     head[u]=ans++;
20 }
21
22 int dfn[MAX+5];
23 int low[MAX+5],num;
24 int sta[MAX+5],top;
25 int scc[MAX+5],sig;
26
27 void dfs(int u)
28 {

```

```

29     dfn[u]=low[u]=++num;
30     sta[top++]=u;
31     for(int i=head[u];~i;i=edge[i].next){
32         int t=edge[i].to;
33         if(dfn[t]==0){
34             dfs(t);
35             low[u]=min(low[u],low[t]);
36         }
37         else if(scc[t]==0){
38             low[u]=min(low[u],low[t]);
39         }
40     }
41     if(dfn[u]==low[u]){
42         sig++;
43         while(1){
44             int j=sta[--top];
45             scc[j]=sig;
46             if(j==u){
47                 break;
48             }
49         }
50     }
51 }
52
53 void tarjan()
54 {
55     sig=0,top=0,num=0;
56     memset(dfn,0,sizeof(dfn));
57     memset(low,0,sizeof(low));
58     memset(sta,0,sizeof(sta));
59     memset(scc,0,sizeof(scc));
60     for(int i=1;i<=n;i++){
61         if(dfn[i]==0){
62             dfs(i);
63         }
64     }
65 }
66
67 int main()
68 {
69     while(~scanf("%d%d",&n,&m)){
70         if(n==0&&m==0){
71             break;
72         }
73         else{
74             memset(head,-1,sizeof(head));
75             ans=0;
76             for(int i=1;i<=m;i++){
77                 int u,v;
78                 cin>>u>>v;
79                 addnode(u,v);
80             }
81             tarjan();
82             if(sig==1){
83                 printf("Yes\n");
84             }
85             else{
86                 printf("No\n");

```

```

87     }
88     }
89 }
90 return 0;
91 }

```

```

1  //CF 427C --Checkposts
2  //链式前向星存图
3  #include<bits/stdc++.h>
4  #define MOD 1000000007
5  #define inf 0x3f3f3f3f
6  typedef long long ll;
7  using namespace std;
8  const int MAXN=300005;
9
10
11 struct node{
12     int to,nxt;
13 }e[MAXN];
14
15 bool vis[MAXN];
16 ll n,m,tot,top,cnt,t;
17 ll ans1,ans2=1;
18
19 int dfn[MAXN],num[MAXN],sum[MAXN],low[MAXN];
20 int siz[MAXN],zhan[MAXN],head[MAXN],val[MAXN];
21
22 void add(int u,int v){
23     e[++tot].to=v;
24     e[tot].nxt=head[u];
25     head[u]=tot;
26 }
27
28 void tarjan(int x){
29     low[x]=dfn[x]=++cnt;
30     zhan[++top]=x;vis[x]=1;
31     for(int i=head[x];i;i=e[i].nxt){
32         int to=e[i].to;
33         if(!dfn[to]) tarjan(to),low[x]=min(low[x],low[to]);
34         else if(vis[to]) low[x]=min(low[x],dfn[to]);
35     }
36     if(dfn[x]==low[x]){
37         ++t;
38         int pre=zhan[top--];
39         sum[t]=val[pre];
40         vis[pre]=0;num[pre]=t;
41         while(pre!=x){
42             pre=zhan[top--];
43             vis[pre]=0;num[pre]=t;
44             sum[t]=min(sum[t],val[pre]);
45         }
46     }
47 }
48
49 void solve(){
50
51     cin>>n;

```

```

52     for(int i=1;i<=n;i++) cin>>val[i];
53     cin>>m;
54     for(int i=1;i<=m;i++){
55         int u,v;
56         cin>>u>>v;
57         add(u,v);
58     }
59     for(int i=1;i<=n;i++) if(!dfn[i]) tarjan(i);
60     for(int i=1;i<=n;i++) if(val[i]==sum[num[i]]) siz[num[i]]++;
61     for(int i=1;i<=t;i++) ans1+=sum[i],ans2=(ans2*siz[i])%MOD;
62     cout<<ans1<<' ' <<ans2%MOD<<endl;
63 }
64
65 int main(){
66     ios::sync_with_stdio(false);
67     cin.tie(0);
68     // int T;
69     // cin>>T;
70     // while(T--){
71         solve();
72     return 0;
73 }

```

杂项

莫队

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define int long long
4  #define ll long long
5  #define fi first
6  #define se second
7  #define vi vector<int>
8  #define vll vector<long long>
9  #define pii pair<int,int>
10 #define endl '\n'
11 #define MOD 1000000007
12 const int N=2e5+5,INF=1e18;
13 int ans,n,T,t,cnt[N],a[N];
14 struct seg{int l,r,id;}q[N],ret[N];
15
16 bool cmp(seg x,seg y){
17     if(x.l/t==y.l/t) return x.r<y.r;
18     return x.l/t<y.l/t;
19 }
20
21 int gcd(int x,int y){
22     return y?gcd(y,x%y):x;
23 }
24
25 void add(int x){
26     ans+=cnt[a[x]];
27     cnt[a[x]]++;

```

```

28 }
29 void del(int x){
30     cnt[a[x]]--;
31     ans-=cnt[a[x]];
32 }
33
34 void solve(){
35     cin>>n>>T;
36     for(int i=1;i<=n;i++) cin>>a[i];
37     for(int i=1;i<=T;i++){
38         cin>>q[i].l>>q[i].r;
39         q[i].id=i;
40     }
41     t=sqrt(n);
42     sort(q+1,q+T+1,cmp);
43     int l=1,r=0;
44     for(int i=1;i<=T;i++){ //先扩再缩
45         while(r<q[i].r) add(++r);
46         while(l>q[i].l) add(--l);
47         while(r>q[i].r) del(r--);
48         while(l<q[i].l) del(l++);
49         int up=ans,down=(r-l+1)*(r-l)>>1;
50         if(r==l){
51             ret[q[i].id].l=0;
52             ret[q[i].id].r=1;
53             continue;
54         }
55         int G=gcd(up,down);
56         up/=G;down/=G;
57         ret[q[i].id].l=up;
58         ret[q[i].id].r=down;
59     }
60     for(int i=1;i<=T;i++) cout<<ret[i].l<<'/'<<ret[i].r<<'\n';
61 }
62
63 signed main(){
64     ios::sync_with_stdio(false);
65     cin.tie(0);
66     int T=1;
67     // cin>>T;
68     while(T--){
69         solve();
70         return 0;
71     }

```

排序

<号表示从小到大

>号表示从大到小

qsort

qsort(数组名 (首地址) , 元素个数, 元素内存大小, int的比较函数)

对二维数组进行比大小

```

1  #include<stdlib.h>
2  //a[num][3]的第三维
3  int cmp(const void *a,const void *b){
4      return ((const int*)a)[2]-((const int*)b)[2];
5  }
6
7  qsort(a,num,sizeof(int[3]),cmp);
8  //-----对等号上面的更改cmp也要更改

```

sort

sort(首地址, 尾地址+1, [cmp函数])

- 传2或3个参数
- 第一个为区间首地址
- 第二个为区间尾地址的下一地址
- 第三个参数不写则缺省为递增排序

sort的cmp为真则说明cmp的第一个参数会在第二个参数的前面, 为假则让第一个参数在第二个参数的后面 (不严谨)

```

1  #include<algorithm>
2
3  bool cmp(int a,int b){
4      return a>b;
5  }
6  sort(num,num+100,cmp);

```

以上从大到小的排序,string类类似

字符数组需要strcmp()函数

```

1  struct node
2  { int a; double b;}arr[100];
3  bool cmp(node x,node y)
4  {
5      if(x.a!=y.a) return x.a<y.a;
6      return x.b>y.b;
7  }
8  sort(arr,arr+100,cmp);

```

以上实现结构体排序,没错浮点数是这样

```

1 struct student
2 {   char name[11]; int age;   double score;
3 }stu[100];
4
5 bool cmp1(student x,student y)
6 {
7     if(fabs(x.score-y.score)>0.000001)
8         return x.score>y.score;
9     if(x.age!=y.age) return x.age<y.age;
10    return strcmp(x.name,y.name)<0;
11 }

```

上述代码是先比较分，再比较年纪，再比较名字字典序

让分高的在前面，年纪小的在前面，名字字典序小的在前面

输入

getline

delim被省略则默认在碰到'\n'后终止输入

<istream>

```

1 istream& getline (char* s, streamsize n );
2 istream& getline (char* s, streamsize n, char delim );
3 //例如
4 cin.getline(name, 256);
5 cin.getline( line, 100, 't' );

```

<string>

重载了四种

```

1 istream& getline (istream& is, string& str, char delim);
2 istream& getline (istream&& is, string& str, char delim);
3 istream& getline (istream& is, string& str);
4 istream& getline (istream&& is, string& str);
5
6 //例如
7 getline(cin, name);
8 getline(cin, name, '#');

```

sscanf

仅在<stdio.h>里面的字符串,处理string要转换为C风格的字符串

```

1 //原型
2 #include <stdio.h>
3 int sscanf(const char *str, const char *format, ...);
4 /*****例子*****/
5 int year, month, day;
6 int converted = sscanf("20191103", "%04d%02d%02d", &year, &month, &day);

```

```

7 //converted=3, year=2019, month=11, day=03
8
9 int converted = sscanf("113.123456789 31.123456789", "%lf %lf", &longitude,
    &latitude);
10 //converted=2, longitude=113.123456789, latitude=31.123457
11 /*****end*****/
12
13 //长度限制
14 char str[32] = "";
15 sscanf("123456abcdedf", "%31[0-9]", str);
16 //str="123456"
17 sscanf("123456abcdedf", "%31[0-9a-z]", str);
18 //str=123456abcdedf
19 sscanf("123456abcdedf586", "%31[^a-z]", str);
20 //str=123456 碰到a~z截止,不再往后读
21 int ret = sscanf("123456abcdedf", "%*[0-9]%31[a-z]", str);
22 //ret=1, str=abcdedf
23 /*表示忽略前面的0~9

```

位运算

```

1 x+y=x^y+2*(x&y)
2
3 01跳变就^1即可
4
5 偶数|1就是加上1

```

GCD

```

1 int gcd(int x,int y){
2     return y?gcd(y,x%y):x;
3 }

```