[hdu3038（种类并查集，推荐）](http://www.cnblogs.com/ziyi--caolu/p/3476968.html)

题目大意：有n次询问，给出a到b区间的总和，问这n次给出的总和中有几次是和前面已近给出的是矛盾的？？

很有意思的一道题目，要是没有做过种类并查集，我肯定会以为这种题目是线段树题目......

思路：我们对给出的区间a,b做以下操作，a-1,b   这样的话，当区间为1--2   3--4的时候，我们可以把两个区间合并成是区间1----4。

在更新的时候，数值小的一定是为根节点，如此我们围绕根节点来更新就可，具体看代码：

#include<iostream>

#include<cstdio>

#include<cstring>

using namespace std;

int father[200005],rank[200005],ans=0;

int find(int x)

{

if(x==father[x])

return x;

int tmp=father[x];

father[x]=find(tmp);

rank[x]+=rank[tmp];

return father[x];

}

void liantong(int x,int y,int k)

{

int xx=find(x);

int yy=find(y);

if(xx==yy)

{

if((rank[y]-rank[x])!=k)

ans++;

return;

}

if(xx>yy)

{

father[xx]=yy;

rank[xx]=rank[y]-rank[x]-k;

}

else

{

father[yy]=xx;

rank[yy]=rank[x]+k-rank[y];

}

}

int main()

{

int n,m;

while(scanf("%d%d",&n,&m)>0)

{

for(int i=0;i<=n;i++)

{

father[i]=i;

rank[i]=0;

}

ans=0;

while(m--)

{

int x,y,k;

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

x--;

liantong(x,y,k);

}

printf("%d\n",ans);

}

return 0;

}

[poj 1733 Parity game(带权并查集)](http://blog.csdn.net/shuangde800/article/details/8010118)

**题目大意：**

有长度为n的0和1组成的字符串， 然后问第L和R位置之间有奇数个1还是偶数个1.

根据这些回答， 判断第几个是错误（和之前有矛盾）的。

#include<iostream>

#include<map>

#include<stdio.h>

using namespace std;

#define MAXN 10010

int f[MAXN],r[MAXN];

//f[]记录父节点，r[]记录与根节点直接1的奇偶性

void init()

{

int i;

for(i=0;i<=10001;i++)

{

f[i]=i;

r[i]=0;

}

}//还是老步骤，初始化并查集

int find(int x)

{

if(f[x]==x)

return x;

int t=find(f[x]);

r[x]=(r[x]+r[f[x]])%2;//这里的话，别当成什么公式吧，可以这样 想，根据当前节点与父节点的关系，以及父节点与根节点的关系，推出当前节点与根节点的关系

f[x]=t;

return f[x];

}//压缩路径

int Union(int x,int y,int d)

{

int a=find(x);

int b=find(y);

if(a==b)

{

if((r[x]+r[y])%2==d)//这个也好理解吧，并查集做了这么多了

return 1;

else return 0;

}

else {

f[a]=b;

r[a]=(r[x]+r[y]+d)%2;//这一步的话，涉及四个节点了，根据其他三个节点之间的关系，觉得第四个节点与根节点的关系

//剩下的还是在压缩路径的过程中解决

return 1;

}

}

int main()

{

int i,j,n,m,a,b,add=0;

int x,y,d;

char s[5];

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

init();

map<int,int> mm;

for(i=1;i<=m;i++)

{

scanf("%d %d %s",&a,&b,s);

a--;

if(mm.find(a)==mm.end())//判定a是否已经映射过了

{

mm[a]=add++;

}

x=mm[a];

if(mm.find(b)==mm.end())

{

mm[b]=add++;

}

y=mm[b];

if(s[0]=='o')

d=1;

else d=0;

if(Union(x,y,d)==0)//其实，测试数据只有一个，所以发现矛盾即可立即退出了

break;

}

printf("%d\n",i-1);

return 0;

}

# [POJ 1733 Parity game(并查集)](http://blog.csdn.net/acm_cxlove/article/details/8083567)

分类： [ACM\_数据结构](http://blog.csdn.net/ACM_cxlove/article/category/1142505)2012-10-17 23:29 1772人阅读 [评论](http://blog.csdn.net/acm_cxlove/article/details/8083567#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/acm_cxlove/article/details/8083567#report)

转载请注明出处，谢谢<http://blog.csdn.net/acm_cxlove/article/details/7854526>    by---cxlove

题目：有一个长度 已知的01串，给出多个条件，[l,r]这个区间中1的个数是奇数还是偶数，问前几个是正确的，没有矛盾

<http://poj.org/problem?id=1733>

开始一点想法都没，虽然知道是并查集，但是不知道怎么做

对于这种区间的，但是稍微转化一下

[l,r]中1个个数可以表示为sum[r]-sum[l-1]，而题目只要求是奇偶，也就确定了sum[r]与sum[l-1]奇偶是否相同，到了这步就简单了，转变为经典模型

范围有点大，按l-1,r离散化一下

#include<iostream>

#include<cstdio>

#include<map>

#include<cstring>

#include<cmath>

#include<vector>

#include<algorithm>

#include<set>

#include<string>

#include<queue>

#define inf 1<<30

#define M 60005

#define N 5005

#define maxn 300005

#define eps 1e-10

#define zero(a) fabs(a)<eps

#define Min(a,b) ((a)<(b)?(a):(b))

#define Max(a,b) ((a)>(b)?(a):(b))

#define pb(a) push\_back(a)

#define mem(a,b) memset(a,b,sizeof(a))

#define LL long long

#define lson step<<1

#define rson step<<1|1

#define MOD 1000000009

**using** **namespace** std;

**struct** Node

{

**int** u,v;

**char** str[10];

}que[N];

**int** a[N\*2],n,q,cnt;

**int** pre[N\*2],r[N\*2];

**int** Bin(**int** x)

{

**int** low=0,high=cnt-1,mid;

**while**(low<=high)

    {

        mid=(low+high)/2;

**if**(a[mid]==x) **return** mid;

**if**(a[mid]<x) low=mid+1;

**else** high=mid-1;

    }

**return** -1;

}

**int** find(**int** x)

{

**if**(x!=pre[x])

    {

**int** f=pre[x];

        pre[x]=find(pre[x]);

        r[x]=r[x]^r[f];

    }

**return** pre[x];

}

**int** main()

{

**while**(scanf("%d",&n)!=EOF)

    {

        scanf("%d",&q);

        cnt=0;

**for**(**int** i=0;i<q;i++)

        {

            scanf("%d%d%s",&que[i].u,&que[i].v,&que[i].str);

            que[i].u--;

            a[cnt++]=que[i].v;a[cnt++]=que[i].u;

        }

        sort(a,a+cnt);

        cnt=unique(a,a+cnt)-a;

**for**(**int** i=0;i<cnt;i++) pre[i]=i,r[i]=0;

**int** ans=0;

**for**(**int** i=0;i<q;i++)

        {

**int** u=Bin(que[i].u),v=Bin(que[i].v);

**int** ra=find(u),rb=find(v);

**if**(ra==rb)

            {

**if**(r[u]==r[v]&&que[i].str[0]=='o') **break**;

**if**(r[u]!=r[v]&&que[i].str[0]=='e') **break**;

                ans++;

            }

**else**

            {

**if**(que[i].str[0]=='o')

                {

                    pre[ra]=rb;

                    r[ra]=r[u]^r[v]^1;

                }

**else**

                {

                    pre[ra]=rb;

                    r[ra]=r[u]^r[v];

                }

                ans++;

            }

        }

        printf("%d\n",ans);

    }

**return** 0;

}