# 2017-2018 ACM-ICPC Northern Eurasia Contest (NEERC 17)

比赛链接

## A. Archery Tournament

### 题意

按时间顺序给定 $n$ 个操作。

1. 操作 $1$ 表示在坐标 $(x,y)$ 上放置一个半径为 $y$ 的靶子。
2. 操作 $2$ 表示在对坐标 $(x,y)$ 进行一次射击，如果命中某个靶子(命中边界不算命中)，则将这个靶子移去。

输出每次射击命中的靶子的编号，保证任何时刻图中现有的靶子不重叠。

### 题解

不难发现，对每次射击询问 $(x,y)$靶子被命中的必要条件是与直线 $x=x_i$ 相交。

又已知任何时刻图中现有的靶子不重叠，于是不难得出结论任何时刻与直线 $x=x_i$ 相交的圆的个数仅有 $\log v$ 个。

于是线段树维护 $x$ 轴上的所有圆的投影即可，时间复杂度 $O(n\log^2 v)$

```
struct cyc{
    int x,y,id;
    bool operator < (const cyc &b)const{
        return id<b.id;
    }
    bool check_in(const pair<int,int> &q)const{
        return 1LL*(x-q.first)*(x-q.first)+1LL*(y-q.second)*(y-
q.second)<1LL*y*y;
    }
};
const int MAXN=2e5+5,MAXM=MAXN*30,MAXV=1e9;
int rt,ls[MAXM],rs[MAXM],node_cnt;
set<cyc> s[MAXM];
cyc cycs[MAXN];
void update(int &k,int nl,int nr,int ql,int qr,cyc c,bool del){
    if(!k)k=++node_cnt;
    if(ql<=nl&&nr<=qr){
        if(del)
        s[k].erase(c);
```

```
        else
        s[k].insert(c);
        return;
    }
    int nm=nl+nr>>1;
    if(nm>=ql)
    update(ls[k],nl,nm,ql,qr,c,del);
    if(nm<qr)
    update(rs[k],nm+1,nr,ql,qr,c,del);
}
int query(int k,int nl,int nr,int pos,pair<int,int> q){
    if(!k)return -1;
    for(set<cyc>::iterator it=s[k].begin();it!=s[k].end();it++){
        if(it->check_in(q))
        return it->id;
    }
    if(nl==nr)return -1;
    int nm=nl+nr>>1;
    if(nm>=pos)
    return query(ls[k],nl,nm,pos,q);
    else
    return query(rs[k],nm+1,nr,pos,q);
}
int main()
{
    int n=read_int();
    _rep(i,1,n){
        int t=read_int(),x=read_int(),y=read_int();
        if(t==1){
            cycs[i]=cyc{x,y,i};
            update(rt,-MAXV,MAXV,max(-MAXV,x-
y),min(MAXV,x+y),cycs[i],false);
        }
        else{
            int ans=query(rt,-MAXV,MAXV,x,make_pair(x,y));
            enter(ans);
            if(ans!=-1)
            update(rt,-MAXV,MAXV,max(-MAXV,cycs[ans].x-
cycs[ans].y),min(MAXV,cycs[ans].x+cycs[ans].y),cycs[ans],true);
        }
    }
    return 0;
}
```