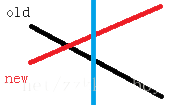
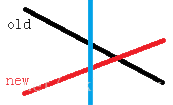
这道题是李超线段树裸题QAQ。说到底，李超线段树就是线段树+标记永久化，所以我们先来说说标记永久化。

我们写线段树区间修改一般都用标记下传的形式。实际上，我们也可以不下传，而是在询问的时候把路径上的标记作用叠加起来（比如区间加，标记不下传，而是在询问的时候累加标记），由此可以发现当路径上的标记无法叠加起来的时候，就不能标记永久化。

上面举的区间加的例子是简单的标记永久化。对于这道题，我们肯定要用一些高级的方法。

对于每个线段树节点，记录“覆盖最广线段” (k,b)(k,b) （顾名思义就是作为最高线段的区间最长的线段，yy一下可以发现询问 x=kx=k 的答案一定在包含 kk 的节点的“覆盖最广线段”中）。然后对于一个节点 oldold ，当有新的线段 newnew 出现的时候，我们需要修改标记：

交点在左边的时候，说明 newnew 是“覆盖最广线段”，先把 oldold 当做新的线段去修改左边，然后将 oldold 修改为 newnew 。

交点在右边的时候，说明 oldold 是“覆盖最广线段”，把 newnew 当做新的线段去修改右边。

因为每次只往一边走，所以效率是 O(nlog22n)O(nlog22n) 。

#include<cstdio>

#include<cctype>

#include<cmath>

#include<algorithm>

using namespace std;

typedef long double DB;

const int maxn=40000,MX=39989,MY=1e9;

int te,tot,lstans;

struct Line {DB k,b;int ID;Line(DB A=0,DB B=0,int C=0) {k=A;b=B;ID=C;}};

Line tr[(maxn<<2)+5];

#define Eoln(x) ((x)==10||(x)==13||(x)==EOF)

inline char readc()

{

static char buf[100000],\*l=buf,\*r=buf;

if (l==r) r=(l=buf)+fread(buf,1,100000,stdin);

if (l==r) return EOF;return \*l++;

}

inline int readi(int &x)

{

int tot=0,f=1;char ch=readc(),lst='+';

while (!isdigit(ch)) {if (ch==EOF) return EOF;lst=ch;ch=readc();}

if (lst=='-') f=-f;

while (isdigit(ch)) tot=(tot<<3)+(tot<<1)+ch-48,ch=readc();

return x=tot\*f,Eoln(ch);

}

inline Line Add(int a,int b,int c,int d)

{

if (a==c) return Line(0,max(b,d),++tot);

DB k=(DB)(d-b)/(c-a);return Line(k,b-k\*a,++tot);

}

inline int fcmp(DB a,DB b) {if (fabs(a-b)<1e-8) return 0;if (a<b) return -1;return 1;}

#define Y(l,x) (l.k\*(x)+l.b)

inline bool cmp(Line a,Line b,int pos)

{

if (!fcmp(Y(a,pos),Y(b,pos))) return a.ID<b.ID;

return fcmp(Y(a,pos),Y(b,pos))>0;

}

#define LS (p<<1)

#define RS (p<<1|1)

void Pushdown(Line x,int l,int r,int p)

{

if (!tr[p].ID) {tr[p]=x;return;}if (cmp(x,tr[p],l)) swap(x,tr[p]);

if (l==r||!fcmp(x.k,tr[p].k)) return;DB pos=(x.b-tr[p].b)/(tr[p].k-x.k);

if (r<pos||pos<l) return;int mid=l+(r-l>>1);

if (pos<=mid) Pushdown(tr[p],l,mid,LS),tr[p]=x; else Pushdown(x,mid+1,r,RS);

}

void Insert(int L,int R,Line x,int l=1,int r=maxn,int p=1)

{

if (R<l||r<L) return;if (L<=l&&r<=R) return Pushdown(x,l,r,p);

int mid=l+(r-l>>1);Insert(L,R,x,l,mid,LS);Insert(L,R,x,mid+1,r,RS);

}

Line Ask(int pos,int l=1,int r=maxn,int p=1)

{

if (l==r) return tr[p];int mid=l+(r-l>>1);Line now;

if (pos<=mid) now=Ask(pos,l,mid,LS); else now=Ask(pos,mid+1,r,RS);

if (cmp(tr[p],now,pos)) now=tr[p];return now;

}

int main()

{

freopen("program.in","r",stdin);

freopen("program.out","w",stdout);

for (readi(te);te;te--)

{

int td,a,b,c,d,x;readi(td);

if (td)

{

readi(a),readi(b),readi(c),readi(d);

a=(a+lstans-1)%MX+1;b=(b+lstans-1)%MY+1;

c=(c+lstans-1)%MX+1;d=(d+lstans-1)%MY+1;

if (a>c) swap(a,c),swap(b,d);Insert(a,c,Add(a,b,c,d));

} else readi(x),printf("%d\n",lstans=Ask((x+lstans-1)%MX+1).ID);

}

return 0;

}