treap

```
#include<cstdio>
#include<iostream>
#include<cstdlib>
#define ls ch[x][0]
#define rs ch[x][1]
#define R register int
using namespace std;
const int N=100010,Inf=0x3f3f3f3f;
inline int g() {
    R ret=0,fix=1; register char ch; while(!isdigit(ch=getchar())) fix=ch=='-'?-1:fix;
    do ret=ret*10+(ch^48); while(isdigit(ch=getchar())); return ret*fix;
}
int n,tot,rt;
int sz[N],ch[N][2],vl[N],dat[N],cnt[N];
inline void upd(int x) {sz[x]=sz[ls]+sz[rs]+cnt[x];}
inline int cre(int v) {R x=++tot; cnt[x]=1,vl[x]=v,dat[x]=rand(),upd(x); return tot;}
inline void rot(int& x,int d) { R y=ch[x][d];
    ch[x][d]=ch[y][d^1]; ch[y][d^1]=x; upd(x),upd(y); x=y;
}
inline void ins(int& x,int v) {
    if(!x) {x=cre(v); return ;}
    if(vl[x]==v) {++cnt[x]; upd(x); return ;} R d=vl[x]<v;
    ins(ch[x][d],v); upd(x); if(dat[ch[x][d]]<dat[x]) rot(x,d);
}
inline void del(int& x,int v) {
    if(!x) return ; if(vl[x]==v) {
        if(cnt[x]>1) --cnt[x]; else {
            if(!ls) x=rs; else if(!rs) x=ls;
            else {R d=dat[ls]>dat[rs]; rot(x,d); del(ch[x][d^1],v);}//看谁大就把谁旋上来,把根旋下去
        }
    } else del(ch[x][vl[x]<v],v); upd(x);
}
inline void build() {srand(100023323); rt=cre(-Inf); ins(rt,Inf);}
inline int getpre(int x,int v) {
    if(!x) return -Inf; if(vl[x]<v) return max(getpre(rs,v),vl[x]);//右边可能没有
    else return getpre(ls,v);
}
inline int getnxt(int x,int v) {
    if(!x) return Inf; if(vl[x]>v) return min(getnxt(ls,v),vl[x]);//同上
    else return getnxt(rs,v);
}
inline int getrk(int x,int v) {
    if(!x) return 0;
    if(vl[x]==v) return sz[ls]+1;
    else if(vl[x]>v) return getrk(ls,v);
    else return sz[ls]+cnt[x]+getrk(rs,v);
}
inline int getvl(int x,int rk) {
    if(!x||!rk) return 0;
    if(rk<=sz[ls]) return getvl(ls,rk);
    else if(rk<=sz[ls]+cnt[x]) return x;
    return getvl(rs,rk-sz[ls]-cnt[x]);
}
signed main() { //freopen("in.in","r",stdin);freopen("out.out","w",stdout);
    n=g(); for(R i=1;i<=n;++i) {
        R k=g(),x=g();
```

```
        if(k==1) ins(rt,x); else if(k==2) del(rt,x);
        else if(k==3) printf("%d\n",getrk(rt,x));
        else if(k==4) printf("%d\n",vl[getvl(rt,x)]);
        else if(k==5) printf("%d\n",getpre(rt,x));
        else printf("%d\n",getnxt(rt,x));
    } //while(1);
}
```

Splay

```
#include<cstdio>
# include<iostream>
<p class="mume-header " id="includeiostream"></p>

# define R register int
<p class="mume-header " id="define-r-register-int"></p>

# define ls (ch[x][0])
<p class="mume-header " id="define-ls-chx0"></p>

# define rs (ch[x][1])
<p class="mume-header " id="define-rs-chx1"></p>

const int N=100005,Inf=0x3f3f3f3f;
using namespace std;
inline int g() {
   R ret=0,fix=1; register char ch; while(!isdigit(ch=getchar())) fix=ch=='-'?-1:fix;
   do ret=ret*10+(ch^48); while(isdigit(ch=getchar())); return ret*fix;
}
int n,tot;
int fa[N],ch[N][2],sz[N],vl[N],cnt[N];
inline int cre(int v) {vl[++tot]=v,sz[tot]=cnt[tot]=1; return tot;}
inline void upd(int x) {sz[x]=sz[ls]+cnt[x]+sz[rs];}
inline void rot(int x) {
   R y=fa[x],d=ch[y][1]==x;
   if(fa[y]) ch[fa[y]][ch[fa[y]][1]==y]=x;
   fa[x]=fa[y]; fa[ch[y][d]=ch[x][d^1]]=y;
   fa[ch[x][d^1]=y]=x; upd(y);
}
int rt;
inline void print(int x) {
   if(!x) return ; print(ls);
   printf("%d\n",vl[x]); print(rs);
}
inline void Splay(int x,int f) {
   while(fa[x]!=f) {
       R y=fa[x]; if(fa[y]!=f)
           rot((ch[y][1]==x)==(ch[fa[y]][1]==y)?y:x); //在不在一条链上
       rot(x);
   } upd(x); if(!f) rt=x;
}
inline void ins(int v) {
   R x=rt; while(1) {
       if(vl[x]==v) {++cnt[x]; break;}
       if(!ch[x][vl[x]<v]) {
           fa[ch[x][vl[x]<v]=cre(v)]=x;
           x=tot; break;
       } x=ch[x][vl[x]<v];
   } Splay(x,0);
}
inline void build() {rt=cre(Inf),ins(-Inf);}
inline int getrk(int v) {
   R x=rt,ret=0; while(1) {
       if(vl[x]==v) {ret+=sz[ls]+1; Splay(x,0); return ret;}
       if(vl[x]<v) ret+=sz[ls]+cnt[x];
       if(!ch[x][vl[x]<v]) {++ret; Splay(x,0); return ret;}
       x=ch[x][vl[x]<v];
   }
```

```
}
inline int getpos(int x,int k) {
   if(!x) return 0;
   if(k<=sz[ls]) return getpos(ls,k);
   if(k<=sz[ls]+cnt[x]) return x;
   return getpos(rs,k-sz[ls]-cnt[x]);
}
inline int getvl(int rk) {R x=getpos(rt,rk); Splay(x,0); return vl[x];}
inline int getmx(int x,int y) {if(!x||!y) return x|y; return vl[x]>vl[y]?x:y;}
inline int ppos(int x,int v) {
   if(!x) return 0; if(vl[x]<v) return getmx(x,ppos(rs,v));
   return ppos(ls,v);
}
inline int getpre(int v) {R x=ppos(rt,v); Splay(x,0); return vl[x];}
inline int getmn(int x,int y) {if(!x||!y) return x|y; return vl[x]<vl[y]?x:y;}
inline int npos(int x,int v) {
   if(!x) return 0; if(v<vl[x]) return getmn(x,npos(ls,v));
   return npos(rs,v);
}
inline int getnxt(int v) {R x=npos(rt,v); Splay(x,0); return vl[x];}
inline void del(int v) {
   Splay(ppos(rt,v),0),Splay(npos(rt,v),rt);
   R& x=ch[ch[rt][1]][0]; if(!(--cnt[x])) x=0; else Splay(x,0);
}
signed main() {
   //freopen("in.in","r",stdin);
   R n=g(); build(); while(n--) {
       R k=g(),x=g();
       if(k==1) ins(x);
       else if(k==2) del(x);
       else if(k==3) printf("%d\n",getrk(x)-1);
       else if(k==4) printf("%d\n",getvl(x+1));
       else if(k==5) printf("%d\n",getpre(x));
       else printf("%d\n",getnxt(x));
   }
   //system("pause"); while(1);
}
```

替罪羊树

```
#include<cstdio>
# include<iostream>
<p class="mume-header " id="includeiostream-1"></p>

# define R register int
<p class="mume-header " id="define-r-register-int-1"></p>

using namespace std;
const double A=0.72,RB=0.53;
const int N=100010;
inline int g() {
   R ret=0,fix=1; register char ch; while(!isdigit(ch=getchar())) fix=ch=='-'?-1:fix;
   do ret=ret*10+(ch^48); while(isdigit(ch=getchar())); return ret*fix;
}
struct node{
   int ls,rs,vl,sz,sum,del;
   #define ls(x) t[x].ls
   #define rs(x) t[x].rs
   #define vl(x) t[x].vl
   #define sz(x) t[x].sz
   #define sum(x) t[x].sum
   #define del(x) t[x].del
}t[N];
int n,rt;
int mem[N],cm,tmp[N],ct;
inline bool ck(int x) {return (double)sz(x)*A<=(double)max(sz(ls(x)),sz(rs(x)));}
inline void dfs(int x) {
   if(!x) return ; dfs(ls(x));
   if(!del(x)) tmp[++ct]=x;
   else mem[++cm]=x;
   dfs(rs(x));
}
inline void build(int& x,int l,int r) {
   R md=l+r>>1; x=tmp[md]; if(l==r) {
       ls(x)=rs(x)=del(x)=0; sz(x)=sum(x)=1; return ;
   } if(l<md) build(ls(x),l,md-1); else ls(x)=0;
   build(rs(x),md+1,r);
   sz(x)=sz(ls(x))+sz(rs(x))+1, sum(x)=sum(ls(x))+sum(rs(x))+1;
}
inline void rebuild(int& x) {
   ct=0; dfs(x); if(ct) build(x,1,ct); else x=0;
}
inline void ins(int& x,int vl) {
   if(!x) {
       x=mem[cm--]; vl(x)=vl,ls(x)=rs(x)=del(x)=0; sz(x)=sum(x)=1; return ;
   } ++sz(x),++sum(x);
   if(vl(x)>=vl) ins(ls(x),vl);
   else ins(rs(x),vl); if(ck(x)) rebuild(x);
}
inline int getrk(int vl) {
   R x=rt; R ret=1; while(x) {
       if(vl(x)>=vl) x=ls(x);
       else {ret+=sz(ls(x))+(del(x)==0); x=rs(x);}
   } return ret;
}
inline int getvl(int rk) {
   R x=rt; while(x) { //cout<<x<<" "<<vl(x)<<endl;
       if(del(x)==0&&sz(ls(x))+1==rk) return vl(x);
```

```
        else {
            if(sz(ls(x))+1>rk) x=ls(x);
            else {
                rk-=sz(ls(x))+(del(x)==0);
                x=rs(x);
            }
        }
    }
}
inline void delrk(int& x,int rk) {
    if(del(x)==0&&sz(ls(x))+1==rk) {del(x)=1; --sz(x); return ;}
    --sz(x); if(sz(ls(x))+(del(x)==0)>=rk) delrk(ls(x),rk);
    else delrk(rs(x),rk-sz(ls(x))-(del(x)==0));
}
inline void delvl(int vl) { R x=getrk(vl); //cerr<<x<<endl;
    delrk(rt,x);
    if(sum(rt)*RB>=sz(rt)) rebuild(rt);
}
signed main() { //freopen("in.in","r",stdin); freopen("out.out","w",stdout);
    n=g(); for(R i=100000;i>=1;--i) mem[++cm]=i;
    while(n--) {
        R k=g(),x=g();
        if(k==1) ins(rt,x); else if(k==2) delvl(x);
        else if(k==3) printf("%d\n",getrk(x));
        else if(k==4) printf("%d\n",getvl(x));
        else if(k==5) printf("%d\n",getvl(getrk(x)-1));
        else if(k==6) printf("%d\n",getvl(getrk(x+1)));
    } //while(1);
}
```

FHQ treap

```
#include<cstdio>
# include<iostream>
<p class="mume-header " id="includeiostream-2"></p>

# include<cstdlib>
<p class="mume-header " id="includecstdlib"></p>

# define R register int
<p class="mume-header " id="define-r-register-int-2"></p>

# define ls(i) ch[i][0]
<p class="mume-header " id="define-lsi-chi0"></p>

# define rs(i) ch[i][1]
<p class="mume-header " id="define-rsi-chi1"></p>

using namespace std;
const int N=500010;
inline int g() {
   R ret=0,fix=1; register char ch; while(!isdigit(ch=getchar())) fix=ch=='-'?-1:fix;
   do ret=ret*10+(ch^48); while(isdigit(ch=getchar())); return ret*fix;
}
int tot;
int ch[N][2],sz[N],vl[N],dat[N];
inline void upd(int x) {sz[x]=sz[ls(x)]+sz[rs(x)]+1;}
inline int cre(int v) {R x=++tot; sz[x]=1,vl[x]=v,dat[x]=rand();return x;}
inline int merge(int x,int y) {
   if(!x||!y) return x+y;
   if(dat[x]<dat[y]) {rs(x)=merge(rs(x),y); upd(x); return x;}
   else {ls(y)=merge(x,ls(y)); upd(y); return y;}
}
inline void split(int o,int v,int& x,int& y) {
   if(!o) {x=y=0; return ;}
   if(vl[o]<=v) x=o,split(rs(o),v,rs(o),y);
   else y=o,split(ls(o),v,x,ls(o)); upd(o);
}
inline int getvl(int x,int rk) {
   while(1) { if(rk<=sz[ls(x)]) x=ls(x);
       else if(rk==sz[ls(x)]+1) return x;
       else rk-=sz[ls(x)]+1,x=rs(x);
   }
}
signed main() { srand(100023323u); R n,x,y,z,rt=0; //freopen("1.in","r",stdin);freopen("out.out","w",stdout)
   n=g(); while(n--) {  //cerr<<n<<" "<<rt<<" "<<endl;
       R k=g(),a=g(); if(k==1) split(rt,a,x,y),rt=merge(merge(x,cre(a)),y);
       else if(k==2) {split(rt,a,x,z),split(x,a-1,x,y); y=merge(ls(y),rs(y)),rt=merge(merge(x,y),z);}
       else if(k==3) split(rt,a-1,x,y),printf("%d\n",sz[x]+1),rt=merge(x,y);
       else if(k==4) printf("%d\n",vl[getvl(rt,a)]);
       else if(k==5) split(rt,a-1,x,y),printf("%d\n",vl[getvl(x,sz[x])]),rt=merge(x,y);
       else if(k==6) split(rt,a,x,y),printf("%d\n",vl[getvl(y,1)]),rt=merge(x,y);
   } while(1);
}
```