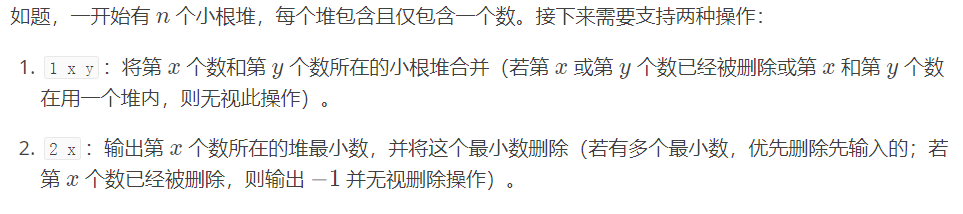
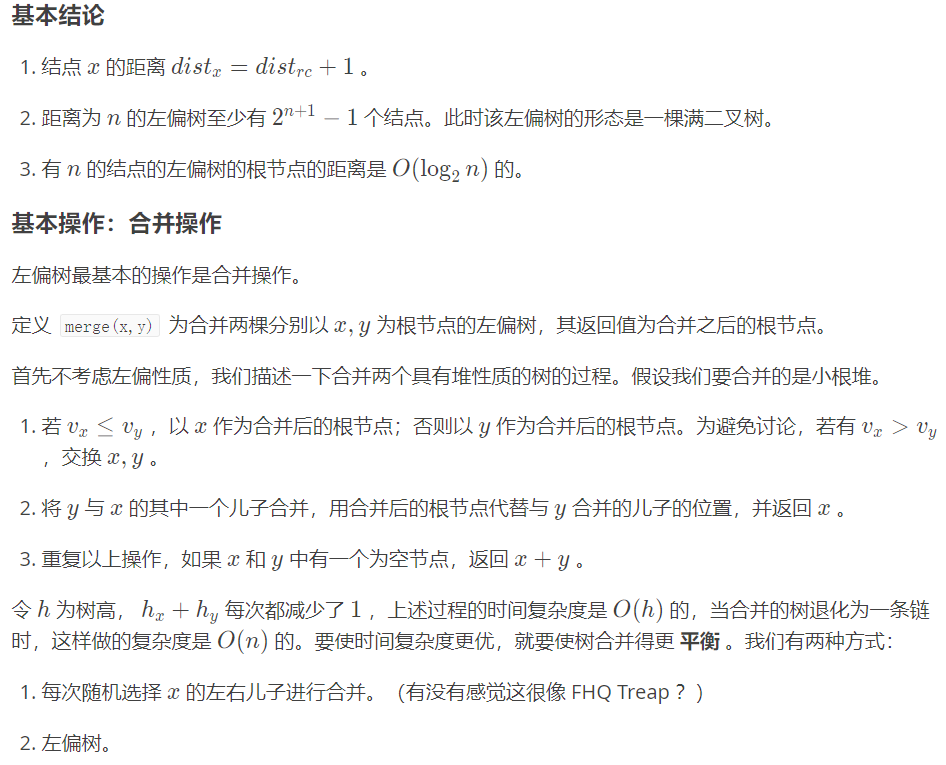
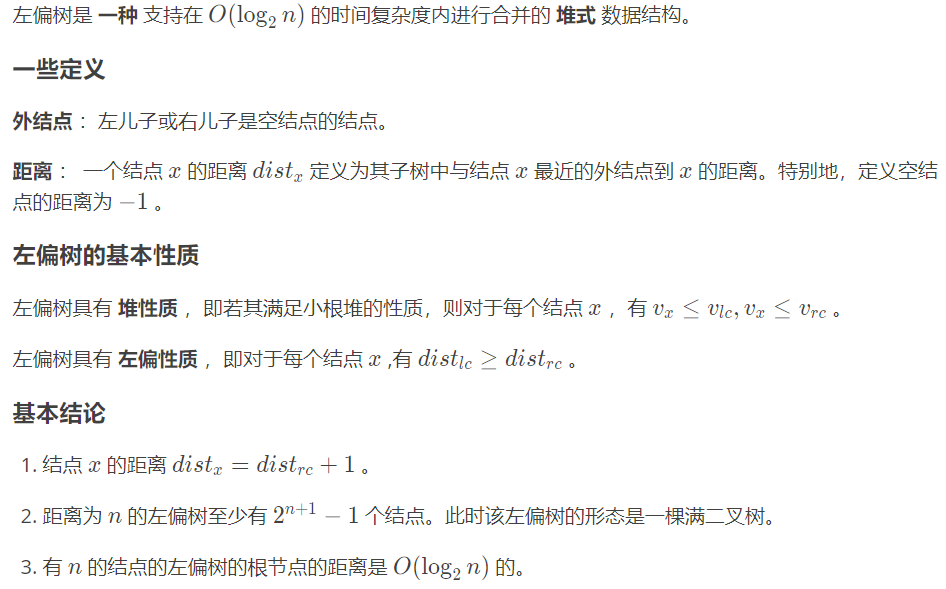
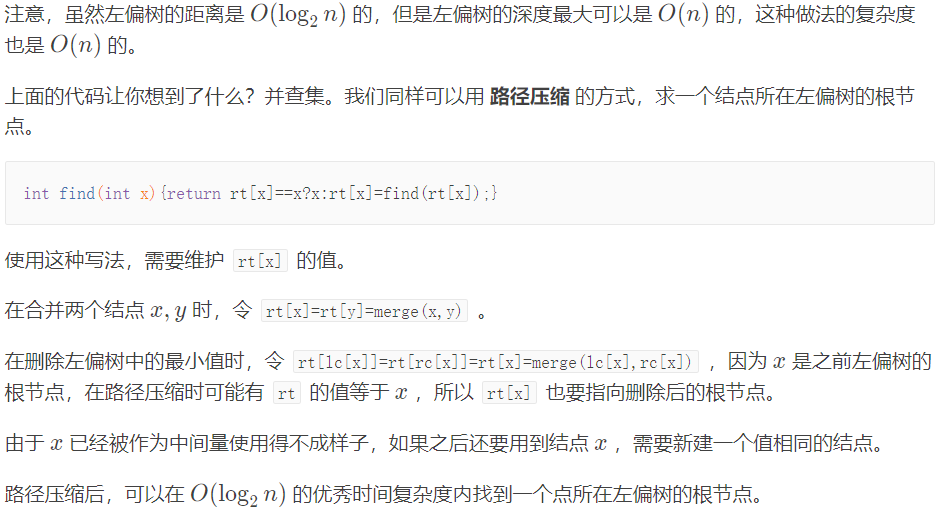
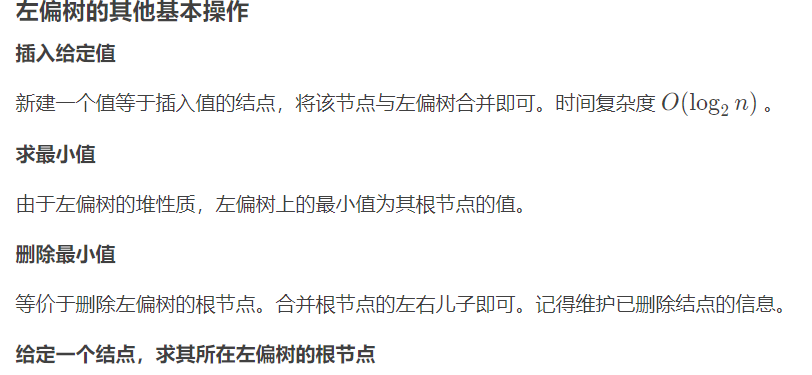
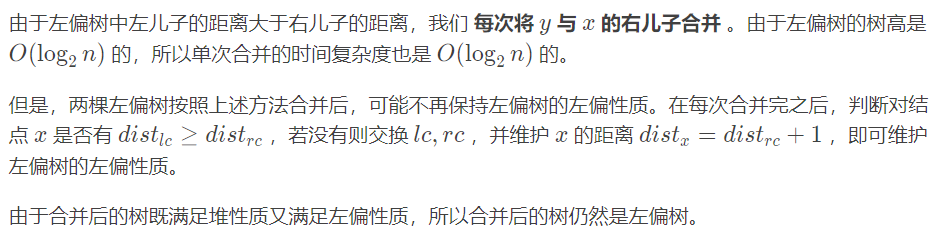
luoguP3377 【模板】左偏树（可并堆）







#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 200005

using namespace std;

ll ans,n,m,prt[mn],v[mn],

ls[mn],rs[mn],d[mn];

bool bj[mn];//这个是用来判断是否节点已经被删除的（此题特有）

ll gf(ll x)

{

if(prt[x]==x)return x;

prt[x]=gf(prt[x]);

return prt[x];

}

ll merge(ll x,ll y)

{

if(!x||!y)return x+y;

if(v[y]<v[x]||v[x]==v[y]&&y<x)swap(x,y);//这个是保证在权值相同的情况下优先删除先输入的那个

rs[x]=merge(rs[x],y);

if(d[ls[x]]<d[rs[x]])swap(ls[x],rs[x]);

d[x]=d[rs[x]]+1;

return x;

}

int main()

{

ll x,y,z,i,j,k;

char ch;

d[0]=-1;//这一行好像是必须添加的

cin>>n>>m;

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

{

scanf("%lld",&v[i]);

prt[i]=i;

}

while(m--)

{

scanf("%lld%lld",&k,&x);

if(k==1)//将第x个数和第y个数所在的小根堆合并

{

scanf("%lld",&y);

if(bj[x]||bj[y])continue;

x=gf(x);y=gf(y);

if(x!=y)prt[x]=prt[y]=merge(x,y);

}

else//输出第x个数所在的堆最小数，并将这个最小数删除

{

if(bj[x]){printf("-1\n");continue;}//已经被删除的节点输出-1

x=gf(x);//得到x所在堆最小值的编号

printf("%lld\n",v[x]);//输出最小值

bj[x]=1;//标记x已经被删除

//下面两行为删除最小值节点x

prt[ls[x]]=prt[rs[x]]=prt[x]=merge(ls[x],rs[x]);

ls[x]=rs[x]=d[x]=0;

}

}

return 0;

}