

堆：

完全二叉树，用数组模拟时，父亲节点的下标是儿子的 $\frac{1}{2}$ （整数部分）（同线段树）
（以小根堆为例）根节点小于儿子节点。

操作：

上浮：某个节点与其父亲比较，如果小于父亲就和父亲交换

下沉：某个节点与两个儿子比较，若大于某个儿子与小的一个儿子交换

插入：插入到最后，然后上浮

弹出：将根与最后一个点交换，再下沉。

左偏树：

我们定义一个节点为外节点，当且仅当这个节点的左子树和右子树中的一个为空节点。一个点的距离，被定义为它子树中离他最近的外节点到这个节点的距离。

性质：

1.左偏树中，任何一个节点的父节点的权值都要小于等于这个节点的权值（堆性质）

2.左偏树中任意一个节点的左儿子的距离一定大于等于右儿子的距离（左偏性质）

推论1. 左偏树中任意一个节点的距离为其右儿子的距离+1

推论2. n个点的左偏树，距离最大为 $\log(n+1)-1$

左偏树的合并：（和treap类似）

1.如果x为空树返回y|y为空返回x

2.val[x]>val[y] swap(x,y)

3.递归rson[x],y

4.检查左右儿子是否符合性质，否则交换

5.更新节点距离（右儿子+1）


模板：

```
#include<bits/stdc++.h>
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<set>
#include<queue>
#include<vector>
#include<cmath>
#include<cstdlib>
#include<map>
#include<stack>
using namespace std;
const int N=100005;
const int inf=0x3f3f3f3f;
void getint(int&num){
    char c;bool flag=0;num=0;
    while((c=getchar())<'0' || c>'9')if(c=='-')flag=1;
    while(c>='0'&& c<='9'){num=num*10+c-48;c=getchar();}
    if(flag) num=-num;
}
int n,T,op,x,y,a[N],ch[N][2],fa[N],d[N];
int merge(int x,int y){
```

```
if ((!x) || (!y)) return x^y;
if (a[x]>a[y]) swap(x,y);
ch[x][1]=merge(ch[x][1],y);
fa[ch[x][1]]=x;
if (d[ch[x][1]]>d[ch[x][0]])
    swap(ch[x][1],ch[x][0]);
d[x]=ch[x][1]?d[ch[x][1]+1]+1:0;
return x;
}
inline int root(int x){
    while(fa[x]!=x)
        x=fa[x];
    return x;
}
inline void _pop(int x){
    fa[ch[x][0]]=ch[x][0];
    fa[ch[x][1]]=ch[x][1];
    merge(ch[x][0],ch[x][1]);

    ch[x][0]=ch[x][1]=fa[x]=d[x]=0,a[x]=-inf;
}
int main(){
    getint(n),getint(T);
    for(int i=1;i=n;i++)
        getint(a[i]),fa[i]=i;
    while(T--){
        getint(op),getint(x);
        if(op==1){
            getint(y);
            if(a[x]==-inf||a[y]==-inf||x==y) continue ;
            int s=root(x),t=root(y);
            merge(s,t);
        }
        else{
            if(a[x]==-inf){puts("-1");continue ;}
            y=root(x),printf("%d\n",a[y]),_pop(y);
        }
    }
}
/*
5 5
1 5 4 2 3
1 1 5
1 2 5
2 2
1 4 2
2 2
*/
```

From:
<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:
<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:famerwzyuki:%E5%B7%A6%E5%81%8F%E6%A0%91&rev=1590331269> 

Last update: **2020/05/24 22:41**