第四题 堆

提交文件:	heap.cpp
输入文件:	heap.in
输出文件:	heap.out
时间空间限制:	2s, 512MB

众所周知，堆是一种可以高效取出最小值的数据结构，其关键操作有上滤和下滤。

具体来说，堆是二叉树的结构，线性的存储在数组 $Heap[1..n]$ 中，节点 $i(1 \le i \le n)$ 的权值为 $Heap[i]$，左儿子为 $2i$，右儿子为 $2i+1$，满足 $Heap[i] \le \min\{Heap[2i], Heap[2i+1]\}$。

对节点 $x(1 \le x \le n)$ 执行上滤操作即不断比较节点 x 以及其父亲的权值，如果更小则交换，C++ 代码如下：

```
void Up(int x):
    while(x>1 && Heap[x]<Heap[x>>1]){
        swap(Heap[x],Heap[x>>1]);
        x>>=1
    }
```

对于节点 $x(1 \le x \le n)$ 执行下滤操作 C++ 代码如下：

```
void Down(int x):
    while(x*2<=n){
        int y=x*2;
        if (y<n && Heap[y+1]<Heap[y]) y++;
        if (Heap[y]>=Heap[x])break;
        swap(Heap[x],Heap[y]);
        x=y;
    }
```

一种时间复杂度是 $O(n)$ 的构建堆的算法是：对于一个 n 个元素的数组 $h[1..n]$，按照从 n 到 1 的顺序对于每个节点执行下滤操作，即可得到一个 n 个节点的堆。

现在有一个 n 的排列 a，以及 q 次询问，询问有两种：

第一种询问给出区间 $[l,r](1 \le l \le r \le n)$ 以及一个数字 $x(1 \le x \le r-l+1)$，对于子序列 $a[l..r]$ 应用上述 $O(n)$ 建立堆的算法之后，求节点 x 的权值。具体来说，对于 $1 \le i \le r-l+1$，记 $b[i] = a[l+i-1]$，将 b 作为权值建立一个 $r-l+1$ 个节点的堆，问堆中节点 x 的权值是多少。

第二种询问同样给出区间 $[l,r](1 \le l \le r \le n)$ 以及一个数字 $x(1 \le x \le n)$，保证 x 是 $a[l..r]$ 中的一个，同样对于子序列 $a[l..r]$ 应用上述算法后，求权值为 x 的节点编号。具体来说，对于 $1 \le i \le r-l+1$，记 $b[i] = a[l+i-1]$，将 b 作为权值建立一个 $r-l+1$ 个节点的堆，问堆中权值为 x 的节点编号是多少。

输入格式

第一行两个整数 n 和 q。

第二行 n 个整数 $a[1], \cdots, a[n]$，表示一个 n 的排列。

接下来 q 行，每行四个整数 ty, l, r, x，其中 ty 表示询问的种类，l, r 表示选择的区间，x 为对应的节点编号或者权值。

输出格式

输出共 q 行，每行输出对应询问的答案。

样例数据

heap.in	heap.out
7 10 5 3 2 1 4 6 7 1 5 7 3 1 3 7 5 1 4 5 2 2 5 7 6 2 6 7 7 1 1 5 2 2 3 3 2 2 7 7 7 2 1 6 1 2 3 7 2	7 7 4 2 2 3 1 1 1 2

数据范围

$n, q \le 1 \times 10^5$

保证 a 是 $1, \cdots, n$ 的排列。

Subtask1(30pts)：$n, q \le 1000$

Subtask2(20pts)：只有询问 1；

Subtask3(20pts)：只有询问 2；

Subtask4(20pts)：$n, q \le 5 \times 10^4$；

Subtask5(10pts)：没有特殊限制。