

# 度度熊学队列

(bear.cpp/.c)

限制：1S 256MB

## 题目描述：

度度熊正在学习双端队列，它对翻转和合并产生了很大的兴趣。初始时有  $N$  个空的双端队列（编号为  $1 \sim N$ ），度度熊的  $Q$  次操作如下。

①  $1\ u\ w\ val$ ：在编号为  $u$  的队列中加入一个权值为  $val$  的元素（ $w=0$  表示加在最前面， $w=1$  表示加在最后面）。

②  $2\ u\ w$ ：询问编号为  $u$  的队列中的某个元素并删除它（ $w=0$  表示询问并操作最前面的元素， $w=1$  表示询问并操作最后面的元素）。

③  $3\ u\ v\ w$ ：把编号为  $v$  的队列“接在”编号为  $u$  的队列的最后面。 $w=0$  表示顺序接（将队列  $v$  的开头和队列  $u$  的结尾连在一起，将队列  $v$  的结尾作为新队列的结尾）， $w=1$  表示逆序接（先将队列  $v$  翻转，再按顺序接在队列  $u$  的后面）。而且在该操作完成后，队列  $v$  被清空。

## 输入：(bear.in)

有多组数据。对于每一组数据，第 1 行都包含两个整数  $N$  和  $Q$ 。接下来有  $Q$  行，每行 3~4 个数，意义如上。 $N \leq 1.5 \times 10^5$ ； $Q \leq 4 \times 10^5$ ； $1 \leq u, v \leq N$ ； $0 \leq w \leq 1$ ； $1 \leq val \leq 10^5$ ；所有数据里  $Q$  的和都不超过  $5 \times 10^5$ 。

## 输出：(bear.out)

对于每组数据的每一个操作②，都输出一行表示答案。如果操作②的队列是空的，则输出 -1 且不执行删除操作。

### 输入样例

```
2 10
1 1 1 23
1 1 0 233
2 1 1
1 2 1 2333
1 2 1 23333
3 1 2 1
2 2 0
2 1 1
2 1 0
2 1 1
```

### 输出样例

```
23
-1
2333
233
23333
```

提示：由于读入过大，建议使用读入优化。一个简单的例子如下。

```
void read(int &x){
    char ch=getchar();x=0;
    for(;ch<'0' || ch>'9';ch=getchar());
    for(;ch>='0'&&ch<='9';ch=getchar()) x=x*10+ch-'0';
}
```

## 题解

### 1 算法设计:

本题描述的就是双端队列，可以使用 `deque` 解决。

- (1) 定义一个 `deque` 数组 `d[]`。
- (2) 判断分别执行 3 种操作，第 2 种操作需要输出。
- (3) 第 3 种情况，由于 `deque` 不支持翻转，因此可以使用反向迭代器控制。

```
if(w)
    d[u].insert(d[u].end(),d[v].rbegin(),d[v].rend());
else
    d[u].insert(d[u].end(),d[v].begin(),d[v].end());
d[v].clear();
```

链表支持翻转和拼接，因此也可以采用链表解决，时间复杂度和空间复杂度更小。

- (1) 定义一个 `list[]`。
- (2) 判断分别执行 3 种操作，第 2 种操作需要输出。
- (3) 第 3 种情况，`list` 支持翻转，拼接函数 `splice` 可以将另一个链表 `v` 拼接到当前链表的 `pos` 位置之前，并自动清空 `v`，且时间复杂度为常数。

```
if(w)
    d[v].reverse();
d[u].splice(d[u].end(),d[v]); //拼接函数 splice 会自动清空 v，时间复杂度为常数
```

### 2. 算法实现

[度度熊学队列 \(deque\) .cpp](#)

[度度熊学队列 \(list\) .cpp](#)