

# 垃圾回收 (garbage)

## 【题目描述】

通常的情况下, 编程语言在管理内存时会进行如下的选择:

- 让用户进行手动内存管理 (C, C++, Rust 等), 这会收获很好的性能, 但是给用户提供了很大的编程负担。
- 使用垃圾回收系统 (Java, Go 等), 这需要维护一个运行时系统, 并且在内存使用和程序性能方面造成了许多不可预测的负担。

尽管存在许多的问题, 目前最通用的自动化内存管理手段始终为 Tracing Garbage Collector。这种做法的最基础的思路是维护对象间的引用关系, 形成一张图, 每次回收时通过扫描引用关系推导出已经无法被访问到的对象, 释放它们占用的内存。而这种传统的做法最大的问题在于维护引用链需要造成很大的开销, 并且随着维护的对象越多, 扫描的代价也会越大。

小 L 是一个喜欢思考的女孩子, 她发现维护 Garbage Collector 是一件非常复杂的事情, 于是她决定考虑一个更简单的模型 (注意它与任何现实中的 GC 规则可能是完全不同的! )。

对于一个  $n$  个点  $m$  条边的无向图, 没有重边自环, 点和边均从 1 开始标号。其中每个节点代表一个占用了一定内存的对象, 每条边对应一个引用关系 (注意这里的引用关系是**无向**的), 程序从第 0 秒开始运行, 在第  $q + 1$  秒结束运行。对于  $i = 1, 2, 3, \dots, q$  的每个时刻  $i$ , 程序会进行以下两种操作之一:

- DELETE  $i$ , 删除边  $(x_i, y_i)$ , 保证不会删除已经被删除的边。
- GC, 进行一次内存回收, 即杀死所有从 1 号点出发不能访问到的点, 释放它们占用的内存。(注意这里对节点的删除不会删除与这些点相连的边)

你可以认为这些操作是被瞬间执行完成的, 在所有操作执行后, 也就是第  $q + 1$  秒, 程序结束, 删除所有剩余的节点 (包括 1 号点)。

第  $i$  个节点占用的内存为  $a_i$ 。现在请你求出整个程序的开销, 即  $\sum_{i=1}^n a_i \cdot t_i$ , 这里  $t_i$  表示第  $i$  个点存活的时间, 在第 0 秒, 所有节点都是存活的。

## 【输入格式】

从文件 *garbage.in* 中读入数据。

输入的第一行是三个正整数  $n, m, q$ , 分别表示对象的个数, 引用关系的个数, 操作的个数。

接下来  $m$  行, 每行 2 个正整数  $x_i, y_i$  表示第  $i$  条引用关系的两个端点。

接下来  $q$  行, 每行为以下两种形式之一, 第  $i$  行表示程序在第  $i$  秒进行的操作:

- 一个字符串 DELETE 和一个正整数  $x$ , 含义见【题目描述】。
- 一个字符串 GC, 含义见【题目描述】。

接下来一行有  $n$  个正整数  $a_1, a_2, \dots, a_n$ , 表示每个对象占用的内存大小。

**【输出格式】**

输出到文件 *garbage.out* 中。

输出一行一个整数表示程序的开销，含义见【题目描述】。

**【样例输入 1】**

见选手目录下的 *garbage/garbage1.in*。

```
6 6 8
1 2
2 3
2 4
1 4
2 5
1 6
GC
DELETE 5
DELETE 3
GC
DELETE 1
GC
DELETE 2
GC
1 2 3 4 5 6
```

**【样例输出 1】**

见选手目录下的 *garbage/garbage1.ans*。

```
149
```

**【样例解释 1】**

在第 4 秒时，节点 5 被删除。

在第 6 秒时，节点 2,3 被删除。

在第 9 秒时，节点 1,4,6 被删除。

因此。程序的开销为  $5 \times 4 + (2 + 3) \times 6 + (1 + 4 + 6) \times 9 = 20 + 30 + 99 = 149$ 。

**【样例 2】**

见选手目录下的 *garbage/garbage2.in* 和 *garbage/garbage2.ans*。

本组数据满足测试点 6 的限制。

**【样例 3】**

见选手目录下的 *garbage/garbage3.in* 和 *garbage/garbage3.ans*。  
本组数据满足测试点 11 的限制。

**【数据规模与约定】**

对于全部数据,  $1 \leq n, m, q \leq 4 \times 10^5$ ,  $1 \leq a_i \leq 10^8$ 。  
具体的数据规模与约定见下表。

| 测试点编号   | $n$                  | $m$                  | $q$                  | 特殊约定       |
|---------|----------------------|----------------------|----------------------|------------|
| 1 ~ 2   | $\leq 500$           | $\leq 500$           | $\leq 500$           | 无          |
| 3 ~ 5   | $\leq 3000$          | $\leq 3000$          | $\leq 3000$          |            |
| 6 ~ 10  | $\leq 5000$          | $\leq 5000$          | $\leq 5000$          |            |
| 11 ~ 14 | $\leq 2 \times 10^5$ | $n - 1$              | $\leq 2 \times 10^5$ | 保证一开始图是一棵树 |
| 15 ~ 16 | $\leq 2 \times 10^5$ | $\leq 2 \times 10^5$ | $\leq 2 \times 10^5$ | 无          |
| 17 ~ 20 | $\leq 4 \times 10^5$ | $\leq 4 \times 10^5$ | $\leq 4 \times 10^5$ |            |