DS #2

|  |  |  |  |
| --- | --- | --- | --- |
| 题目名称 | 最小值 | 最大值 | 前k大 |
| 输入文件名 | min.in | max.in | kth.in |
| 输出文件名 | min.out | max.out | kth.out |
| 每个测试点时限 | 1s | 1s | 1s |
| 测试点数目 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 |
| 内存限制 | 256M | 256M | 256M |
| 是否有部分分 | 否 | 否 | 否 |
| 题目类型 | 传统 | 传统 | 传统 |

注意：代码长度限制均为64K，不开O2。时限可能根据标程的实际运行时间修改，为标程运行时间2倍以上。

1 **最小值**(min.c/cpp/pas)

1.1 **题目描述**

有一棵 n 个点的有根树，每个结点拥有一个点权，现在有 m 次操作，每次操作为如下三种之一：

* 1 x y：点 x 的权值改为 y
* 2 x：查询以 x 为根的子树中，权值最小的结点的权值
* 3 x：更改整棵树的根为 x

1.2 **输入格式**

第一行两个整数 n, m , 分别表示树的大小和操作数。

接下来n行,每行两个整数f, v,第i+1行的两个数表示点i的父亲和点i的权。保证f < i，1号结点为根节点，输入的 f 将为 0。

接下来 m 行，每行以1/2/3作为第一个数，表示操作种类，接下来一或两个整数，格式同题目描述，表示一次操作。

1.3 **输出格式**

对于每一个2操作，输出一行一个整数，表示询问的答案。

1.4 **样例输入**

3 4

0 1

1 3

1 2

1 1 4

2 1

3 3

2 1

1.5 **样例输出**

2

3

1.6 **数据范围与约定**

对于30％的数据，n , m<= 103。

对于另外20％的数据，不带换根操作。

对于100％的数据，n, m<=105，给出的所有数绝对值不超过109，且保证操作均合法。

2 **最大值** (max.c/cpp/pas)

2.1 **题目描述**

有一棵 n 个点的树，边按照1~n-1标号，每条边拥有一个边权，现在有 m 次操作，每次操作为如下三种之一：

* 1 x y：边 x 的权值改为 y
* 2 x y：将点 x 到点 y 路径上的所有边权值变成相反数
* 3 x y：查询点 x 到点 y 路径上的最大边权

2.2 **输入格式**

第一行为两个整数n ，m，表示序列长度和操作次数。

接下来 n-1 行，每行三个数 a, b, v，表示 a、b 之间有一条权值为 v 的边。边按照标号1~n-1顺序给出。

接下来 m 行，每行以1/2/3作为第一个数，表示操作种类，接下来两个整数，格式同题目描述，表示一次操作。

2.3 **输出格式**

对于每一个3操作，输出一行一个整数，表示询问的答案。

2.4 **样例输入**

3 3

1 2 3

2 3 2

3 1 3

2 1 3

3 1 3

2.5 **样例输出**

3

-2

2.6 **数据范围与约定**

对于30％的数据，n , m<= 103。

对于100％的数据，n, m<=105，给出的所有数绝对值不超过109，且保证操作均合法（如不会出现右端点小于左端点的情况）。

3 **前k大**(kth.c/cpp/pas)

3.1 **题目描述**

你有一个长为 n 的序列，你需要在长度处于 L 到 R 之间的所有子段中，找到子段和前 K 大的所有子段，并且输出和。

3.2 **输入格式**

第一行四个数 n, K, L, R，含义如题目所述。

接下来一行 n 个数，表示这个序列。

3.3 **输出格式**

对于每一个询问，输出一行表示答案。

3.4 **样例输入**

4 3 2 3

3 2 -6 8

3.5 **样例输出**

11

3 .6 **数据范围与约定**

对于 30% 的数据，满足 n, K <= 103 ；

对于 100% 的数据，满足 n, K <= 105，且序列的数Ai满足 -103<=Ai<=103。且保证一定存在第 K 大。