

NOIP 2019 模拟赛 Contest 18

diamond_duke

题目名称	视频	图论	排队
可执行文件名	video	graph	queue
输入文件名	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出
时间限制	1s	1s	1s
内存限制	512MB	512MB	512MB
子任务个数	4	4	4
题目类型	传统型	传统型	传统型

请注意： 评测时开启 O2 优化和 C++11 编译选项，栈空间限制同空间限制。

1 视频

1.1 Problem Statement

小 D 正在研究视频。

考虑一个 n 帧的视频。该视频在播出时，每一帧的持续时间为 d 秒，即第 i 帧将在第 $(i-1) \cdot d$ 秒时开始播出，并持续播出 d 秒。

为了播放视频，小 D 需要先对帧进行解码：一个帧只有在被解码后才可以播出，且第 i 帧解码所需时间为 a_i 。

然而，帧的解码之间具有依赖关系。所有帧被分为三种：I,P,B，分别表示帧内编码帧、前向预测编码帧，以及双向预测内插编码帧。他们之间的依赖关系如下：

- I 帧的解码不需要依赖任何帧；
- P 帧的解码需要依赖其前面最后一个 I 或 P 帧；
- B 帧需要同时依赖其前面最后一个 I 或 P 帧，以及后面第一个 I 或 P 帧。

小 D 希望按照播出顺序进行解码。唯一的例外是，如果 B 帧依赖的某个帧还没有解码，那么应当先对那个帧进行解码，再回来对 B 帧进行解码。

由于内存十分宝贵，因此小 D 希望使用的内存尽量少。具体而言，在某个帧进行解码或被依赖时，其必须位于内存中。**当某帧开始被播放时，它必须位于内存中，然后会被从内存复制到播放器的寄存器中。**如果某个帧不再被需要，那么其可以被从内存中丢弃。值得一提的是，小 D **不允许**解码一个帧多次，被丢弃的帧也无法重新放回内存。此外，小 D **可以在任意时间开始进行解码的过程。**

小 D 希望知道最小的内存大小为多少帧，才能在每秒至多同时解码一帧的情况下**不卡顿**地播放视频（即每当某帧要被播放时，它都已经在内存中被解码完了）。

但是小 D 并不会，请你帮帮他。

1.2 Input Format

从标准输入读入数据。

第一行两个正整数 n, d ，表示视频中帧的个数，以及每一帧的持续时间。

第二行一个长度为 n 的，仅由 I,P,B 组成的字符串，其中第 i 个字符表示第 i 帧的类型。

第三行 n 个空格隔开的整数，其中第 i 个数 a_i 表示第 i 帧需要的解码时间。

1.3 Output Format

向标准输出输出答案。

输出一行一个整数，表示最小需要的内存大小。

1.4 Sample 1

1.4.1 Input

```
12 8
IBBPBBPPIBBP
100 5 3 15 4 3 12 7 25 3 2 8
```

1.4.2 Output

```
4
```

1.4.3 Explanation

我没有想到一个绝妙的解释，因此这里地方太小，写不下。

1.5 Sample 2

见下发文件 `video/video2.in` 与 `video/video2.ans`。

1.6 Sample 3

见下发文件 `video/video3.in` 与 `video/video3.ans`。

1.7 Constraints

对于所有测试数据， $1 \leq n \leq 3 \times 10^5$ ， $1 \leq a_i, d \leq 10^9$ ，保证每一帧依赖的帧存在。

- 子任务 1 (20 分): $n \leq 500$;
- 子任务 2 (30 分): $n \leq 5000$;
- 子任务 3 (30 分): $n \leq 10^5$;
- 子任务 4 (20 分): 无特殊限制。

2 图论

2.1 Problem Statement

小 D 正在研究图论。

小 D 画了一张 n 个点的无向图，初始为空，节点被依次编号为 $1, 2, \dots, n$ 。

然后，小 D 写出了 m 对点 (x, y) ，表示他不喜欢这些点对。

小 D 接下来给出了 q 条边。小 D 会依次考虑其中每条边：如果这条边加入会引起某条小 D 不喜欢的点对可以相互到达，那么小 D 不会加入这条边，否则会加入这条边。

小 D 想对于每条边，求出这条边是否会被加入。

但是小 D 并不会，请你帮帮他。

2.2 Input Format

从标准输入读入数据。

第一行三个正整数 n, m, q ，分别表示图中的点数、小 D 不喜欢的点对个数，以及边数。

接下来 m 行，每行两个整数 x, y ，表示小 D 不喜欢点对 (x, y) 。

接下来 q 行，每行两个整数 u, v ，表示小 D 加入的一条边。

2.3 Output Format

向标准输出输出答案。

输出一行一个长度为 q 的 0/1 字符串，其中第 i 位为 1 表示这条边被加入了，否则没有加入。

2.4 Sample 1

2.4.1 Input

```
3 1 3
1 2
1 2
2 3
1 3
```

2.4.2 Output

```
010
```

2.5 Sample 2

见下发文件 `graph/graph2.in` 与 `graph/graph2.ans`。

2.6 Sample 3

见下发文件 `graph/graph3.in` 与 `graph/graph3.ans`。

2.7 Constraints

对于所有测试数据, $1 \leq n, m, q \leq 10^5$, $1 \leq x, y \leq n$, $x \neq y$, $1 \leq u, v \leq n$, $u \neq v$ 。

- 子任务 1 (20 分): $n, m, q \leq 2000$;
- 子任务 2 (20 分): $n, m, q \leq 5000$;
- 子任务 3 (30 分): $n, m, q \leq 5 \times 10^4$;
- 子任务 4 (30 分): 无特殊限制。

3 排队

3.1 Problem Statement

小 D 正在去排队吃饭的路上。

走在小 D 前面的共有 n 个人，小 D 将这些人按照从前到后的顺序依次编号为 $1, 2, \dots, n$ 。

这些人会按照先后顺序依次来到窗口前。但是由于队伍太长了，他们可能并不会按照顺序进行排队：

每当一个人过来时，他会选择一个同班同学，并站在他的前面或者后面。同时，为了防止被太多人发现自己插队的事实，他会选择一个不会让太多人向后移动的位置。

特别地，如果有多个这样的位置，那么他会站在其中最靠前的那个；如果找不到这样的位置，那么他会站在队伍的末尾。

小 D 经过一些调查，得到了第 i 个人的班级为 c_i ，而可以接受的移动的人数为 a_i 。即，第 i 个人不能接受有超过 a_i 个人因为他的插队行为而向后移动。

小 D 想要知道，在轮到他到窗口前时，正在排队的 n 的人的顺序是什么。

但是小 D 并不会，请你帮帮他。

3.2 Input Format

从标准输入读入数据。

第一行一个正整数 n ，表示走在小 D 前面的人数。

接下来 n 行，每行两个整数 c_i, a_i ，表示第 i 个人的班级以及可以接受的移动人数。

3.3 Output Format

向标准输出输出答案。

输出一行 n 个整数，其中第 i 个数字表示排在第 i 位的人的编号。

3.4 Sample 1

3.4.1 Input

```
8
1 0
2 1
2 2
1 1
```

3 2
1 3
2 3
2 5

3.4.2 Output

1 3 8 2 7 6 4 5

3.5 Sample 2

见下发文件 `queue/queue2.in` 与 `queue/queue2.ans`。

3.6 Sample 3

见下发文件 `queue/queue3.in` 与 `queue/queue3.ans`。

3.7 Constraints

对于所有测试数据, $1 \leq n \leq 5 \times 10^5$, $1 \leq c_i \leq n$, $0 \leq a_i \leq n$ 。

- 子任务 1 (25 分): $n \leq 5000$;
- 子任务 2 (25 分): $n \leq 5 \times 10^4$;
- 子任务 3 (30 分): $n \leq 2 \times 10^5$;
- 子任务 4 (20 分): 无特殊限制。