

第 36 届全国信息学奥林匹克竞赛

CCF NOI 2019

模拟试题

竞赛时间：2019 年 1 月 28 日 8:00–12:00

| | | | |
|---------|------------|------------|------------------|
| 题目名称 | 余弦 | 小 Cat 与小鲜肉 | 最短路 |
| 目录 | cosine | sjy | shortest |
| 可执行文件名 | cosine | sjy | N/A |
| 输入文件名 | cosine.in | sjy.in | shortest1~10.in |
| 输出文件名 | cosine.out | sjy.out | shortest1~10.out |
| 每个测试点时限 | 2 秒 | 1 秒 | N/A |
| 内存限制 | 512MB | 512MB | N/A |
| 测试点数目 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 |
| 是否有部分分 | 否 | 否 | 是 |
| 题目类型 | 传统型 | 传统型 | 提交答案 |
| 是否有附加文件 | 是 | 是 | 是 |

提交源程序须加后缀

| | | | |
|--------------|------------|---------|-----|
| 对于 Pascal 语言 | cosine.pas | sjy.pas | N/A |
| 对于 C 语言 | cosine.c | sjy.c | N/A |
| 对于 C++ 语言 | cosine.cpp | sjy.cpp | N/A |

注意：最终测试时，所有编译命令均不打开任何优化开关。

余弦

【问题描述】

给定长度为 N 的实数序列 $A_i (1 \leq i \leq N)$, 你需要在数列上进行两类操作:

1. 把 $l \leq i \leq r$ 中的每个 A_i 加上实数 v 。
2. 求 $l \leq i \leq r$ 中 $\cos(A_i)$ 的和。

【输入格式】

从文件 **cosine.in** 中读入数据。

输入包含多组数据, 最开头的正整数 $T (T \leq 5)$ 指明了数据组数。对于每组数据:

第一行 2 个整数 N, M , 表示数列长度与操作个数。

第二行 N 个实数 $A_i (|A_i| \leq 100)$, 小数点后最多有 3 位小数。

接下来 M 行每行形如“1 $l r v$ ”或“2 $l r$ ”。前者表示一个第 1 类操作, 后者表示一个第 2 类操作。其中 $l, r (1 \leq l \leq r \leq N)$ 是整数, 而 $v (|v| \leq 100)$ 是实数, 小数点后最多有 3 位小数。

【输出格式】

输出到文件 **cosine.out** 中。

对于每组数据, 先输出一行 "Case #k:", 其中 k 是该组数据的编号, 从 1 开始。

对于每组数据中的每个第 2 类操作, 输出一个实数表示答案, 保留 3 位小数。

【样例输入 1】

```
3
3 3
0 -2 -6
1 1 2 -7
2 1 2
2 2 3
5 4
4 2 4 -0 -4
1 1 4 2
2 4 5
1 3 5 5
2 2 3
7 3
0.123 -19.002 -57.507 34.434 -80.886 -17.115 69.843
2 2 7
1 2 7 16.341
```

2 1 3

【样例输出 1】

Case #1:

-0.157

0.049

Case #2:

-1.070

-0.649

Case #3:

1.854

-0.842

【样例输入输出 2】

见选手目录下的 *cosine/cosine.in* 与 *cosine/cosine.ans*。

【数据规模与约定】

对于 20% 的数据， $1 \leq N, M \leq 1000$ 。

对于另外 20% 的数据，没有 1 类操作。

对于另外 20% 的数据，1 类操作均在 2 类操作之前。

对于 100% 的数据， $1 \leq N, M \leq 200000$ 。

小 Cat 与小鲜肉

【问题描述】

P 校某宿舍人才辈出，其舍长图书馆男神因被偷拍侧身照而在网络上一票走红。小鲜肉 SJY 是小 Cat Rainbow 的好朋友，他也是该宿舍的一员。作为一名著名的程序设计师，小鲜肉 SJY 不但注重萌萌哒的外表，还掌握了无数的黑科技。有一天，SJY 制造了一块比特板，这个比特板有 2^N 个比特元，编号为 $0 \sim 2^N - 1$ 。每个比特元有一个饱和值 T ，可以接收一个 $[0, 2^M)$ 之间的整数作为输入信号，并产生整数 P 作为固定的输出信号。当编号为 i 的比特元接收到输入信号 j 时，将生成 $W(i, j)$ 枚比特币。

相似的比特元之间还会产生叠加效果。如果两个比特元的编号 a 和 b 在二进制下只有一位不同，并且两个比特元中的至少一个接收到的输入信号不小于其饱和值时，这两个比特元将额外生成 $P_a \text{ xor } P_b$ 枚比特币。SJY 希望给每个比特元适当的输入信号，使比特板生成的比特币总数尽量多。SJY 认为这个问题太简单了，作为一名小鲜肉，比赚钱更重要的是出去赢得无数妹子的目光，所以他把这个问题交给你解决。

【输入格式】

从文件 *sjy.in* 中读入数据。

第一行两个整数 N, M 。

第二行 2^N 个整数 T_i ，表示每个比特元的饱和值。

第三行 2^N 个整数 P_i ，表示每个比特元的固定输出信号。

接下来 2^N 行每行 2^M 个整数 $W(i, j)$ ，比特币是虚拟货币，所以 $W(i, j)$ 可能是负数。

【输出格式】

输出到文件 *sjy.out* 中。

一个整数，表示最多能生成的比特币数。

【样例输入 1】

```
3 2
0 1 1 3 3 0 3 3
4 8 8 7 0 9 2 9
-9 -8 3 2
-9 -6 4 1
-6 -8 -5 3
3 -1 -4 -1
-6 -5 1 10
-10 7 3 -10
```

-3 -10 -4 -5
-2 -1 -9 1

【样例输出 1】

133

【样例说明 1】

输入信号依次为 2 2 3 0 3 1 0 3。

【样例输入输出 2】

见选手目录下的 *sjy/sjy.in* 与 *sjy/sjy.ans*。

【数据规模与约定】

对于 20% 的数据， $1 \leq n \leq 3, 1 \leq m \leq 2$ 。

对于另外 20% 的数据， $T_i = 0$ 或 2^m 。

对于另外 20% 的数据， $m = 1$ 。

对于 100% 的数据， $1 \leq n \leq 8, 1 \leq m \leq 8, 0 \leq T_i \leq 2^m, 0 \leq P_i, |W(i,j)| \leq 1024$ 。

最短路

【问题描述】

给定一个节点 1 和节点 N 连通的正权无向图 G ，请你删除不超过 K 条边，使得节点 1 和节点 N 仍然连通的同时，这两点之间的最短路尽可能长。

【输入格式】

本题为提交答案试题，输入文件 `shortest1.in~shortest10.in` 已经在试题目录下。

输入文件 `shortest*.in` 的第一行包含三个正整数 N ， M 和 K 。其中 N 表示节点数， M 表示边数，节点的编号由 1 至 N ，边的编号由 1 至 M 。接下来 M 行，每行三个正整数 u ， v 和 w ，表示有一条连接节点 u 和节点 v 的边，权值为 w 。

【输出格式】

输出文件 `shortest*.out` 的第一行应该包含一个非负整数 T ($T \leq K$)，表示需要删掉的边数。

接下来 T 行，每行一个 1 到 M 之间的整数 x ，表示删掉输入中的第 x 条边。你需要保证这 T 个整数互不相同。

【样例输入】

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

【样例输出】

```
1
3
```

【数据规模与约定】

对于每个测试点，设有评分四个参数 s_1, s_2, s_3, s_4 。假设你的方案的最短路为 ans 。

如果你没有输出，或者输出不合法，或者最短路不存在，得 0 分。

如果最短路存在，得 1 分。

如果 $ans \geq s_1$ ，得 3 分。

如果 $ans \geq s_2$ ，得 5 分。

如果 $ans \geq s_3$ ，得 8 分。

如果 $\text{ans} = \text{s4}$ ，得 10 分。

如果 $\text{ans} > \text{s4}$ ，得 12 分。

取满足条件的分数中的最高得分为该测试点你的得分。

在你的目录下有一个名为 `checker` 的程序可以用来检查你的输出，你可以在命令提示符中使用以下命令来检查你的输出：

`checker N`

其中 N 为测试点的编号，例如，要测试第 3 个测试点可以使用

`checker 3`

该程序会检测你的输出方案是否合法。如果方案合法，程序还会给出该方案的最短路的长度值。