

SDSY NOIP 模拟赛

2023 年 10 月 12 日 14:00-18:00

一. 题目概况

中文题目名称	老师的考验	天气	图	朋友
英文题目与子目录名	trail	weather	graph	friends
输入文件名	trail.in	weather.in	graph.in	friends.in
输出文件名	trail.out	weather.out	graph.out	friends.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统
运行内存上限	512	512	512	512

注意事项:

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、C/C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
- 3、统一评测时采用的机器配置为：Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz，内存 8G，上述时限以此配置为准。

1. 老师的考验

(`trail.cpp/c/pas`)

【题目描述】

小花在高一数学课上学习了幂函数的相关知识，老师决定考验她的速算能力，老师希望小花可以快速计算出 m 个 2^n 的十进制首位数字，可是小花并没有这么强的速算能力，作为一个训练有素的竞赛选手，请你帮助小花完成老师的考验。

【输入格式】

从`trial.in`中读取数据。

第一行一个整数 m ，意义见题目描述。

以下 m 行，每行一个整数 n ，意义见题目描述。

【输出格式】

输出至文件`trail.out`中。

共 m 行，每行一个整数，为题目要求的首位数字。

【样例 1 输入】

1
64

【样例 1 输出】

1

【样例 1 解释】

只有一次询问， $n = 64$ ， 2^{64} 的十进制数约为 1.8×10^{19} ，首位为1，故输出1。

【样例 2 输入】

2
362
375

【样例 2 输出】

9
7

【样例 2 解释】

有两次询问。

第一次询问 $n = 362$ ， 2^{362} 的十进制数约为 9.39×10^{108} ，首位为9，故输出9。

第二次询问 $n = 375$ ， 2^{375} 的十进制数约为 7.70×10^{112} ，首位为7，故输出7。

【数据范围】

测试点编号	$n \leq$	特殊性质
1	64	无
2	10^6	保证 $n = w * 10^z w, z \in N^*$
3~6	10^6	无
7~10	10^{12}	无
对于所有的测试点, $m \leq 10$		

2. 天气

(weather.cpp/c/pas)

【题目描述】

熊猫王国有 n 个城市，城市之间由 m 条有向道路连接。第 i 条道路从 u_i 号城市走到 v_i 号城市。

熊猫王国财政赤字严重，道路都年久失修，所以，每一条道路的通过所需体力，每天都会随着天气发生变化。每一天，第 i 条道路有 50% 的概率通过所需体力为 a_i ，有 50% 的概率通过所需体力为 b_i 。

熊猫王国地域辽阔，不同的时间与地区，天气往往并不相同，所以，同一天的不同道路的通过所需体力、不同日子里道路的通过所需体力，都是完全独立随机的。

小猫的家在 1 号城市，她想沿着这些道路，去 n 号城市找小熊玩。每一天的早晨，小猫都会收听天气预报，得知当天每一条道路的通过所需体力，然后再决定当天沿着哪一条道路走。小猫做决定时知道的信息，只有当天的每一条道路的通过所需体力情况。每一天，小猫都必须恰好走过一条道路。

小猫想知道，如果小猫采取最优策略，从 1 号城市走到 n 号城市所需的体力之和的期望值的最小值是多少。

【输入格式】

第一行两个整数 n 和 m ，表示城市数量和道路数量。

接下来 m 行，每行四个正整数 u_i, v_i, a_i, b_i ，表示每一条道路的起点、终点，以及两个可能的通过所需体力。

【输出格式】

一行一个实数，表示从 1 号城市走到 n 号城市所需的体力之和的期望值的最小值。

设你的答案为 a ，标程的答案为 b ，你的答案会被判定为正确，当且仅当

$$\frac{|a - b|}{\max(1, b)} < 10^{-5}$$

【样例输入】

```
3 3
1 2 3 5
2 3 5 8
1 3 2 100
```

【样例输出】

```
6.25
```

【样例解释】

如果小猫当前位于 2 号城市，那么，接下来行走的道路只能是第 2 条，期

望还要消耗 $\frac{5+8}{2} = 6.5$ 的体力。

如果小猫当前位于 1 号城市，那么：

- 有 50% 的概率，第 3 条道路的通过所需体力为 2，此时直接花费 2 的体力，从 1 号城市走到 3 号城市。
- 另 50% 的概率，第 3 条道路的通过所需体力为 100，此时只能走第 1 条道路，花费期望 $\frac{3+5}{2} = 4$ 的体力后到达 2 号城市，到达 2 号城市后还要再花费期望 6.5 的体力才能走到 3 号城市。

所以，从 1 号城市出发，期望所需体力之和为 $\frac{2 + (4+6.5)}{2} = 6.25$ 。

【数据范围】

对于所有数据，有：

- $1 \leq n, m \leq 10^5$
- $1 \leq a_i, b_i \leq 10^4$
- 保证存在至少一条从 1 号城市走到 n 号城市的路径。
- 可能存在两条道路 i, j 满足 $u_i = u_j$ 且 $v_i = v_j$ （起点和终点都相同）。
- 可能存在一条道路 i ，使得 $a_i = b_i$ （该边通过所需体力不会变化）。
- 不存在一条道路 i 使得 $u_i = v_i$ （起点和终点相同）。

以下为各 Subtask 分别的限制：

Subtask 编号	n	m	其他限制	分值
1	≤ 5	≤ 20		20
2			对于每一条道路 i 都有 $a_i = b_i$	15
3			图是一张 DAG（有向无环图）	15
4	≤ 1000	≤ 1000		30
5				20

3. 图

(graph.cpp/c/pas)

【题目描述】

现有一有向图 G ，节点编号为 $1 \dots n$ ，你可以在其中任意加一条边 $i \rightarrow j$ (其中 $2 \leq i, j \leq n-1$)，边权为 $k \times |i - j|$ 。在此条件下求 1 到 n 的最短路。

【输入格式】

第一行三个正整数 n, m, k 以空格隔开。

之后 m 行每行三个正整数 s, t, w ，表示 s 到 t 有一条边权为 w 的有向边。

【输出格式】

如果有解，输出一行一个正整数，表示答案；如果无解，输出 -1 。

【输出格式】

如果有解，输出一行一个正整数，表示答案；如果无解，输出 -1 。

【样例输入】

```
5 3 12
1 2 10
2 3 10
4 5 10
```

【样例输出】

```
42
```

【样例解释】

添加一条边 3 到 4 ，边权为 $12 \times |3 - 4| = 12$ ，最短路为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 = 10 + 10 + 12 + 10 = 42$ 。

【数据范围】

对于 30% 的数据， $3 \leq n \leq 300$ ；

对于 60% 的数据， $3 \leq n \leq 5000$ ；

对于 100% 的数据， $3 \leq n \leq 100000, 1 \leq m \leq 200000, 1 \leq w, k \leq 10^9$
不保证数据中没有自环和重边。

4 朋友

(friends.cpp/c/pas)

【问题描述】

Friends 是人类的信仰。

现在我们有 N 个天蓝怪和 M 个 Friends。每个天蓝怪和 Friends 都有自己的朋友值，同时给出以下规定：

- 1、如果两个天蓝怪拥有朋友值 a, b ，并且 $a \text{ xor } b \equiv 1(\text{mod } 2)$ ，那么这两个天蓝怪是朋友。
- 2、如果两个 Friends 拥有朋友值 a, b ，并且 $a \text{ xor } b \equiv 0(\text{mod } 2)$ 或者 a or b 在二进制下有奇数个一，那么两个 Friends 是朋友。

如果我们同时告诉你天蓝怪和 Friends 之间的朋友关系，我们希望找到最多数量的天蓝怪和 Friends，使得她们彼此之间都是朋友。问你最多能找到多少天蓝怪和 Friends 满足这个条件。

【输入格式】

第一行一个数 T ，代表该测试点内的数据组数，接下来 T 组数据。

对于每组数据，第一行三个整数 N, M, K 。

第二行 N 个整数代表每个天蓝怪的朋友值。

第三行 M 个整数代表每个 Friends 的朋友值。

接下来 K 行每行两个整数 i, j ，代表第 i 个天蓝怪和第 j 个 Friends 是朋友。

【输出格式】

对于每组数据，输出一个整数代表答案。

【样例输入】

```
1
2 4 7
1 2
2 6 5 4
1 1
1 2
1 3
2 1
2 2
2 3
2 4
```

【样例输出】

5

【数据规模与约定】

对于其中30%的数据, $N = 0, M \leq 100$;
对于其中50%的数据, $N \leq 10, M \leq 100$;
对于其中10%的数据, $N \leq 5, M \leq 1000$;
对于其中10%的数据, $N \leq 5, M \leq 1500$;
对于100%的数据, $K \leq N \times M$, 朋友值在 2^{30} 以内。