

全锅青少年信息学奥林匹克联赛模拟赛

Boom Round 1

Day⁻² ∈ ℝ

Tx_lcy,boom_fst

时间：2023 年 6 月 27 日 8:30 ~ 11:45

题目名称	图	随机	子串 I
题目类型	传统型	传统型	非传统型
目录	graph	random	substring
可执行文件名	graph	random	substring
输入文件名	graph.in	random.in	substring.in
输出文件名	graph.out	random.out	substring.out
每个测试点时限	1.0 ~ 6.0 秒	1.5 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
子任务数目	50	38	33
子任务是否等分	是	否	是

提交源程序文件名

对于 C++ 语言	graph.cpp	random.cpp	substring.cpp
-----------	-----------	------------	---------------

编译选项

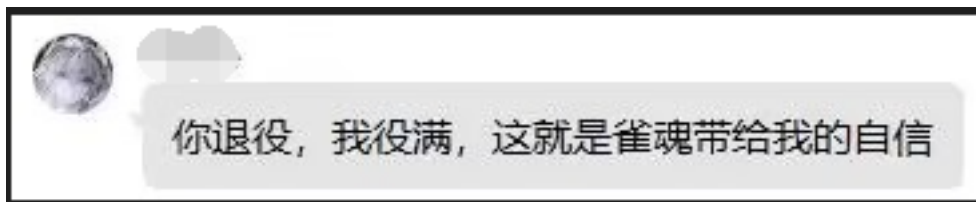
对于 C++ 语言	-std=c++14 -lm -O2
-----------	--------------------

注意事项（请仔细阅读）

1. 请不要在比赛时大声嘴巴。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
4. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
5. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 评测在最新公布的 NOI Linux 下进行，使用 LemonLime 评测。
9. 因为测评时间太长，T1 仅保证在 std 时限 1.5 倍左右，其余均保证在 2 倍以上。

图 (graph)

【题目背景】



【题目描述】

一姬在一个 n 个点和 m 条边无向图中迷路了，她不知道她现在在哪里。每个点上有一个宝玉，一姬要收集 k 个宝玉才能缔结契约，走出这个无向图。图中被访问的点不能再访问第二次，经过每条边需要一定的时间，求所需的最大时间是多少？

注：走到的点宝玉必须要取走。收集到 k 个宝玉必须离开无向图。

【输入格式】

从文件 `graph.in` 中读入数据。

第一行输入包括一个整数 T ，表示测试案例的数量。

每个测试用例开始都有一行三个整数 n, m, k ，分别表示迷宫中的房间数和道路数，以及他需要收集的宝石数。

接下来的 m 行中的每一行都包含三个整数 u, v, t ，表示一条从 u 到 v 的无向边，通过需要 t 分钟。无重边，无自环。

【输出格式】

输出到文件 `graph.out` 中。

输出共 T 行，其中第 i 行表示所需的最大时间。若无解输出 -1 。

【样例 1 输入】

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

```
8 1 2 2
9 2 3 3
10 4 5 5
```

【样例 1 输出】

```
1 11
2 -1
```

【样例 1 解释】

第一组数据：2->1->4

第二组数据：一姬无法收集到 4 个宝玉。

【样例 2】

见选手目录下的 *graph/graph2.in* 与 *graph/graph2.ans*。

【数据范围】

对于所有数据，保证 $1 \leq T \leq 5, 2 \leq n \leq 5000, 1 \leq m \leq 5000, 2 \leq k \leq 6, 1 \leq t \leq 10^6$ 。

1 ~ 35 号测试点时限为 1s，36 ~ 50 号测试点时限为 6s。

各测试点的附加限制如下表所示：

测试点编号	$n, m \leq$	$k \leq$
1 ~ 5	40	6
6 ~ 11	5000	2
12 ~ 17	1000	3
18 ~ 23	1000	4
24 ~ 29	1000	5
30 ~ 35	1000	6
36 ~ 40	3000	6
41 ~ 50	5000	6

随机 (random)

【题目背景】



你退役，他役满，我被飞，我们都有光明的未来。

【题目描述】

给定一张 n 个点， m 条边的图，你可以进行如下两种操作：

1. 选择一条边并删掉，这个操作代价为 a 。
2. 选择一个点并删掉与之关联的所有边，这个操作代价为 b 。

你希望以最小的代价删掉图中所有的边。请你输出这个最小代价。

【输入格式】

从文件 *random.in* 中读入数据。

第一行一个正整数 T ，表示数据组数。

对于每组数据：

第一行 n, m, a, b ，代表图的点数和边数，和删掉边和点的代价。

接下来 m 行，每行两个正整数 $u, v (1 \leq u, v \leq n)$ ，代表图上的一条边。图中可能存在重边和自环。

【输出格式】

输出到文件 *random.out* 中。

对于每组数据，一行一个正整数表示答案。

【样例 1 输入】

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

```
8 4 5 3 4
9 1 4
10 1 3
11 1 2
12 2 4
13 3 4
14 5 6 6 7
15 1 2
16 3 4
17 4 5
18 5 2
19 3 3
20 1 4
```

【样例 1 输出】

```
1 6
2 8
3 20
```

【数据范围】

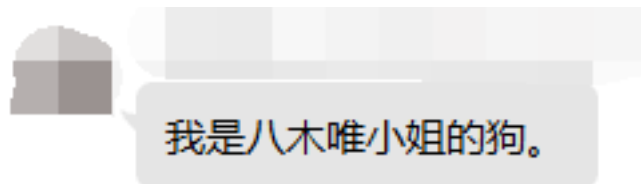
对于 100% 的数据， $1 \leq T \leq 10, 1 \leq n, m \leq 40, 1 \leq a, b \leq 10^8$ 。

各测试点的附加限制如下表所示：

测试点编号	$n, m \leq$	特殊性质
1 ~ 7	20	每个点 3 分
8 ~ 16	30	每个点 3 分
17 ~ 24	40	保证数据随机，每个点 3 分
25 ~ 38	40	每个点 2 分

子串 I(*substring*)

【题目背景】



【题目描述】

八木唯给你了一个 n ，输出一个 01 序列，使得其中的本质不同子串最多。

我们定义两个子串不同，当且仅当有这两个子串长度不一样或者长度一样且有任意一位不一样。

子串的定义：原字符串中连续的一段字符组成的字符串。

【输入格式】

从文件 *substring.in* 中读入数据。

输入第一行是一个正整数 n 。

【输出格式】

输出到文件 *substring.out* 中。

输出第一行是一个长度为 n 的 01 字符串 a 。

【样例 1 输入】

1 4

【样例 1 输出】

1 0011

【样例 1 解释】

不同子串数量为 8，分别是 0, 1, 00, 01, 11, 001, 011, 0011。可以证明没有更多的方案，但这不是唯一方案。

【样例 2】

见选手目录下的 *substring/substring2.in* 与 *substring/substring2.ans*。

【样例 3】

见选手目录下的 *substring/substring3.in* 与 *substring/substring3.ans*。

【数据范围】

共有 33 个测试点。

对于 1 ~ 17 测试点 i 的数据, $n = 2^{i-1}$ 。

对于 18 ~ 33 号测试点 i 的数据, $n = 2^{i-17} + (i - 17) - 1$ 。

若最优答案为 X , 你的答案为 Y , 该测试点得分为 S , 评分如下:

范围	得分
$Y < \min(X - 100, X/2)$	0
$\min(X - 100, X/2) \leq Y < X - 100$	$0.2 \cdot S$
$X - 100 \leq Y < X - 40$	$0.3 \cdot S$
$X - 40 \leq Y < X - 10$	$0.5 \cdot S$
$X - 10 \leq Y < X - 4$	$0.7 \cdot S$
$X - 4 \leq Y < X$	$0.8 \cdot S$
$Y = X$	S

注:

1. 因为 lemon 技术受限, 无法显示小数, 所以本题每个数据点 10 分。

对于所有测试点, 将会按每 3 ~ 5 个数据点捆绑为一组, 共 10 组。

保证按照数据梯度捆绑。

得分均取包中得分最小值。

2. 本题下发的 checker 并非最终评测使用的 checker, 但是效果一样, 望周知。