~~**注意：测试的时候按点测试，子任务分布仅供参考**~~

A 回文块

时空限制：1s 128MB

文件名

palindromic.in/palindromic.out/palindromic.cpp

题目描述

给出一个只包含小写字母字符串，要求你将它划分成尽可能多的小块，使得这些小块的字符串构成回文序列。

例如：对于字符串 `abcab`，将他分成 `ab+c+ab` 或者 `abcab` 就是构成回文串的划分方法，`abc+ab` 则不是。

对于任何串，显然不做任何拆分得到的就是回文序列，该回文序列长度为 1 。现在求拆分得到的回文序列长度的最大值。

输入格式

第一行输入一个整数 T 表示数据组数。

接下来的 T 行，每行输入一个字符串，代表你需要处理的字符串，保证该字符串只包含小写字母。

输出格式

输出 T 行，对于每个输入的字符串，输出一行包含一个整数 x，表示该字符串最多能分解成 x 个小块，使得这些小块构成回文串。

样例 #1

样例输入 #1

```
4
bonobo
deleted
racecar
racecars
```

样例输出 #1

```
3
5
7
1
```

提示

对于 100% 的数据, 有 $1 \leq T \leq 10$ 。设 L 为单个字符串的长度, 则 $1 \leq L \leq 10^6$ 。

- 子任务 $1(15\%)$: $1 \leq L \leq 30$;
- 子任务 $2(20\%)$: $1 \leq L \leq 300$;
- 子任务 $3(20\%)$: $1 \leq L \leq 10^4$;
- 子任务 $4(45\%)$: 无特殊限制。

B 灯泡

时空限制：2s 256MB

文件名

sure.in/sure.out/sure.cpp

题目描述

现在有 n 个A类灯泡和 n 个B类灯泡，每个灯泡都有各自的权值。

我们将这些灯泡分为 n 组，每组包含一个来自A类的灯泡和一个来自B类的灯泡。

你可以从中选取任意个灯泡，每选取一个灯泡需要花费 1 的代价。

在你选取完之后，系统会随机在A类和B类中选择一个类型，并点亮那一类的所有灯泡。你选取的每个点亮的灯泡会给你带来等于它权值的收益。

现在请你合理选取灯泡，以最大化可能的最小收益。你只需要求出来这个收益即可。

输入格式

第一行一个正整数 n ，表示灯泡的组数。

接下来 n 行每行两个空格隔开的实数 A_i, B_i。分别表示属于这组的A灯泡和B灯泡的权值。输入的实数不会超过四位小数。

输出格式

输出最大化的最小可能收益，请输出到小数点后恰好四位。

样例 #1

样例输入 #1

```
4
1.4 3.7
1.2 2
1.6 1.4
1.9 1.5
```

样例输出 #1

```
0.5000
```

提示

样例解释

最优策略是选择第一组的B灯泡和第三组的A灯泡和第四组的A灯泡：

- 如果B类灯泡被点亮，收益是 $3.7-3=0.7$。
- 如果A类灯泡被点亮，收益是 $1.6+1.9-3=0.5$ 。

最小可能收益是 0.5。

数据范围 对于所有测试点, 有 $1.0 \leq A_i, B_i \leq 1000.0$ 。

- 子任务 $1(20\%)$: 有 $1 \leq n \leq 10$;
- 子任务 $2(40\%)$: 有 $1 \leq n \leq 1000$;
- 子任务 $3(40\%)$: 有 $1 \leq n \leq 100000$ 。

C 袋鼠

时空限制：1s 128MB

文件名

kangaroo.in/kangaroo.out/kangaroo.cpp

题目描述

有一个园子，里面有 n 个草丛排成一排，标号 $1 \sim n$，有一个袋鼠，从 s 出发，每次跳一步跳到一个其他的草丛，经过每个草丛恰好一次，最终到达 t。显然他会跳跃 $n-1$次为了不被人类发现，袋鼠每次跳跃的方向必须与前一次不同。

具体地，如果他现在在 now，他是从 $prev$ 跳跃一次到达 now 的，然后他跳跃一次到达 $next$：

- 那么如果 $prev < now$，就必须有 $next < now$；
- 如果 $now < prev$，就必须有 $now < next$。

问从 s 到 t 的方案数模 10^9+7的结果。

两个路线不同，当且仅当草丛被访问的顺序不同。

保证至少有一种方案初始时可以往任意方向跳。

输入格式

一行三个整数 n, s, t。

输出格式

一行一个整数，代表答案。

样例 #1

样例输入 #1

```
4 2 3
```

样例输出 #1

```
2
```

提示

对于 40% 的数据，$2 \le n \le 100$

对于 100% 的数据，$2 \le n \le 2 \times 10^3$，$1 \le s, t \le n$

D 换乘

时空限制：2s 256MB

文件名

commuter.in/commuter.out/commuter.cpp

题目描述

JOI 君住在一个有 N 个车站的城市。车站编号为 1 至N。编号为 1 至 M 的有 M 条铁路。铁路 i（$1 \leq i \leq M$）双向连接 A_i 站和 B_i 站，票价为 C_i 日元。
JOI 君住在 S 站附近，去 T 站附近的 IOI 高中。他打算买一张连接这两个站的通勤票。当他购买通勤票时，他需要选择一条成本最低的 S 站和 T 站之间的路线。使用此通勤票，他可以沿任何方向乘坐所选路线中包含的任何铁路，而无需额外费用。
JOI 君经常去 U 站和 V 站附近的书店。因此，他想买一张通勤票，这样从 U 站到 V 站的费用可以降到最低。
当他从 U 站移动到 V 站时，他首先选择了一条从 U 站到 V 站的路线，那么他需要支付的车费是：

- 0 日元：如果铁路 i 包含在他购买的通勤票时选择的路线中，或者，
- C_i 日元：如果铁路 i 不包含在他购买通勤票时选择的路线中。

上述票价的总和就是从 U 站到 V 站的成本。
他想知道如果他在购买通勤票时选择合适的路线，从 U 站到 V 站的最低成本。

输入格式

第一行包含两个空格分隔的整数 N，M。这意味着 JOI 君居住的城市有 N 个车站和 M 条铁路。第二行包含两个空格分隔的整数 S，T。这意味着 JOI 君计划购买从 S 站到 T 站的通勤票。第三行包含两个空格分隔的整数 U，V。这意味着 JOI 君想要最小化从站 U 到站 V 的成本。接下来的 M 行的第 i 行包含三个空格分隔的整数 A_i，B_i，C_i。铁路 i 双向连接 A_i 站和 B_i 站，票价为 C_i 日元。

输出格式

如果他在购买通勤票时选择了适当的路线，则输出应包含一个整数为从站 U 到站 V 的最小成本。

样例 #1

样例输入 #1

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

样例输出 #1

```
2
```

样例 #2

样例输入 #2

```
6 5
1 2
3 6
1 2 1000000000
2 3 1000000000
3 4 1000000000
4 5 1000000000
5 6 1000000000
```

样例输出 #2

```
3000000000
```

样例 #3

样例输入 #3

```
8 8
5 7
6 8
1 2 2
2 3 3
3 4 4
1 4 1
1 5 5
2 6 6
3 7 7
4 8 8
```

样例输出 #3

```
15
```

样例 #4

样例输入 #4

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

样例输出 #4

```
0
```

样例 #5

样例输入 #5

```
10 15
6 8
7 9
2 7 12
8 10 17
1 3 1
3 8 14
5 7 15
2 3 7
1 10 14
3 6 12
1 5 10
8 9 1
2 9 7
1 4 1
1 8 1
2 4 7
5 6 16
```

样例输出 #5

```
19
```

提示

数据规模与约定

对于 100% 的数据，
$2 \leq N \leq 10^5$，$1 \leq M \leq 2 \times 10^5$，$1 \leq S \leq N$，$1 \leq T \leq N$，$1 \leq U \leq N$，$1 \leq V \leq N$，$S \neq T$，$U \neq V$，$S \neq U$ 或 $T \neq V$，$1 \leq A_i \leq B_i \leq N$（$1 \leq i \leq M$）。对于每 $1 \leq i < j \leq M$，$A_i \neq A_j$ 或 $B_i \neq B_j$，$1 \leq C_i \leq 10^9$（$1 \leq i \leq M$）。JOI 君可以从任何车站移动到任何其他车站乘坐铁路。

- Subtask 1（12 points）：$S = U$。
- Subtask 2（15 points）：从 S 站到 T 站有一条成本最低的唯一路线。
- Subtask 3（24 points）：$N \leq 300$。
- Subtask 4（45 points）：没有额外的限制。

样例说明

对于样例 1：JOI 君在购买通勤票时只能选择一条路线：1 号站 → 2 号站 →3 号站 → 5 号站 → 6 号站。为了尽量减少从 1 号站到 4 号站的成本，他选择以下路线：1 站 → 2 站 → 3 站 → 5 站 → 4 站。选择这条路线时，他需要支付的车费为。

- 2 日元用于车站 4 和车站 5 的铁路 5，以及，
- 0 日元：其他铁路。

对于样例 2：JOI 君从 3 号站移动到 6 号站时，不使用通勤票。