

# NOI 模拟赛

2025 年 6 月 21 日

|        |          |         |            |
|--------|----------|---------|------------|
| 题目名称   | 城市漫游     | 线段的划分   | 凯撒密码       |
| 题目类型   | 传统型      | 传统型     | 传统型        |
| 可执行文件名 | city     | seg     | caesar     |
| 输入文件名  | city.in  | seg.in  | caesar.in  |
| 输出文件名  | city.out | seg.out | caesar.out |
| 时间限制   | 1.0 秒    | 1.0 秒   | 2.0 秒      |
| 内存限制   | 512 MB   | 512 MB  | 512 MB     |
| 测试点数目  | 20       | 捆绑测试    | 捆绑测试       |

提交源程序文件名

|        |          |         |            |
|--------|----------|---------|------------|
| 对于 C++ | city.cpp | seg.cpp | caesar.cpp |
|--------|----------|---------|------------|

编译选项

|        |                |
|--------|----------------|
| 对于 C++ | -O2 -std=c++14 |
|--------|----------------|

## 注意事项（请仔细阅读）

1. 选手提交的源程序请放置在以题目对应文件名命名的子文件夹下。
2. 这套模拟赛的题目不一定按照难度顺序排列!!!
3. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
4. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
5. 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。
6. 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。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。

## 城市漫游 (city)

### 【题目描述】

小 P 来到了一个 City，这个 City 有  $n$  个街口，且有  $m$  条道路连接这  $n$  个街口，保证所有街口之间都互相连通。

现在小 P 想要从街口 1 去到街口  $n$ ，每 1 个单位时间小 P 可以通过一条道路从一个街口去往另一个街口。然而，不怀好意的小 G 打算阻挠小 P，具体来说，小 G 会进行恰好一次行动：选择任意一个时间点，以及任意一条道路将其彻底封锁（若小 P 此刻正在某条道路上，则这条道路不可被封锁），在这个时间点之后小 P 都无法通过这条道路。

小 P 可以在道路被封锁后立即获得消息，现在假设小 P 和小 G 都按最优策略行动，而小 P 想要最小化从街口 1 去到街口  $n$  花费的时间，而小 G 想要最大化时间，请你求出小 P 最后到底花了多少单位的时间。

### 【输入格式】

从文件 *city.in* 中读入数据。

第一行输入两个整数表示  $n, m$ 。

接下来  $m$  行，每行两个整数  $u, v$  表示有一条道路连接街口  $u$  和街口  $v$ 。

### 【输出格式】

输出到文件 *city.out* 中。

输出一个整数，表示最终花费的单位时间。如果小 P 无论如何也无法从街口 1 去到街口  $n$  则输出  $-1$ 。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
2
```

**【样例 2】**

见下发文件中的 *city/ex\_city2.in* 与 *city/ex\_city2.out*。

**【测试点约束】**

对于 30% 的数据，满足  $n, m \leq 2000$ 。

对于另外 10% 的数据，满足  $m = n - 1$ 。

对于另外 20% 的数据，满足  $m = n$ 。

对于全部数据，满足  $2 \leq n \leq 2 \times 10^5$ ， $n - 1 \leq m \leq \min(\frac{n(n-1)}{2}, 2 \times 10^5)$ 。保证不存在重复的道路。保证所有道路连接着不同的街口。

## 线段的划分 (seg)

### 【题目描述】

小 G 在一个数轴上画了  $n$  条线段，第  $i$  个线段可以用一个  $(L_i, R_i)$  表示线段的左右端点。由于小 G 有强迫症，所以他画出的线段满足性质：

- 每个线段的端点都在整数位置且所有线段的端点两两不同。
- 对于任意两条不同的线段，要么没有交集，要么其中一个完全包含另一个。

现在小 G 提出了一个问题：如果给每个线段选择一个**非空**的子线段满足所有子线段两两无交（**端点可以相交**），所有子线段长度和最大是多少。

换句话说，你需要给每个线段  $(L_i, R_i)$  选择一个  $(l_i, r_i)$  满足  $L_i \leq l_i < r_i \leq R_i$  且任意一对  $(l_i, r_i)$  无交，最大化  $\sum_{i=1}^n (r_i - l_i)$ 。

### 【输入格式】

从文件 *seg.in* 中读入数据。

第一行一个整数  $n$  表示线段数量。

接下来  $n$  行，每行两个整数  $L_i, R_i$  表示线段左右端点。

### 【输出格式】

输出到文件 *seg.out* 中。

输出一行一个整数表示所有子线段长度和的最大值。

### 【样例 1 输入】

```
5
0 8
15 16
9 11
2 4
1 7
```

### 【样例 1 输出】

```
10
```

【样例 1 解释】

选择的子区间序列可能为  $\{(0, 2), (15, 16), (9, 11), (2, 4), (4, 7)\}$ 。

【样例 2】

见下发文件中的 *seg/ex\_seg2.in* 与 *seg/ex\_seg2.out*。

【样例 3】

见下发文件中的 *seg/ex\_seg3.in* 与 *seg/ex\_seg3.out*。

【样例 4】

见下发文件中的 *seg/ex\_seg4.in* 与 *seg/ex\_seg4.out*。

【样例 5】

见下发文件中的 *seg/ex\_seg5.in* 与 *seg/ex\_seg5.out*。

【测试点约束】

本题采用捆绑测试，只有你通过了一个 subtask 中的所有数据才能获得其分数。

对于全部数据，满足  $1 \leq n \leq 5 \times 10^3$ ， $0 \leq L_i < R_i \leq 10^9$ ，所有线段端点两两不同且任意一对线段要么不存在交集要么完全包含。

| 子任务 | 分值 | $n \leq$ | 特殊性质   |
|-----|----|----------|--|
| 1   | 25 | 10       | 所有 $R_i \leq 20$                                     |
| 2   | 15 |          |  |
| 3   | 15 | 50       | 无  |
| 4   | 15 | 400      |  |
| 5   | 10 | 5000     | $\forall 1 \leq i < n, L_i < L_{i+1}, R_i > R_{i+1}$ |
| 6   | 20 |          | 无  |

## 凯撒密码 (caesar)

### 【题目描述】

凯撒密码是一种最简单且最广为人知的加密技术，具体来说凯撒密码将所有字母循环向后移动固定步长  $d$ ，例如当  $d = 1$  时，有  $b \rightarrow a, c \rightarrow b, \dots, a \rightarrow z$ 。而如果限制字母表的大小为  $k (k \leq 26)$  使得字符串仅保留前  $k$  个小写字母，那么循环也会同时缩小，如  $k = 3, d = 1$  时，有  $b \rightarrow a, c \rightarrow b, a \rightarrow c$ 。

小 P 是一个凯撒密码的狂热研究者，于是小 G 送给小 P 一个字符串  $s$  来帮助他进行研究，字符串  $s$  限制字母表大小为  $k$ 。这时，小 G 有  $q$  个问题，每个问题形如：

小 G 给出一个区间  $[l, r]$ ，小 P 可以选择区间  $[l, r]$  的一个子区间  $[l', r'] (l \leq l' \leq r' \leq r)$  并取出子串  $s[l', r']$ ，并再选择一个正整数  $d$ ，对  $s[l', r']$  做步长为  $d$  的凯撒密码加密。在进行恰好一次上述对字符串  $s$  的加密后，小 G 想最大化  $s$  中最长的回文串的长度，所以他想让小 P 合理地选择子区间  $[l', r']$  和步长  $d$  使得长度最大化。注意：每次询问相互独立。

### 【输入格式】

从文件 `caesar.in` 中读入数据。

第一行输入一个字符串  $s$ 。

第二行输入两个整数  $k, q$ 。

接下来  $q$  行，每行两个整数  $l, r$  表示一次询问的区间。

### 【输出格式】

输出到文件 `caesar.out` 中。

输出  $q$  行，每行一个整数表示一次询问的最长回文串长度。

### 【样例 1 输入】

```
bcac
3 5
3 4
1 4
2 2
4 4
4 4
```

**【样例 1 输出】**

```
4
4
3
3
3
```

**【样例 1 解释】**

对第 1, 2 次询问, 选择子区间  $[3, 4]$  和  $d = 2$  可得  $s = \text{bccb}$ , 最长回文串长度为 4。

对第 3, 4, 5 次询问, 选择子区间  $[3, 3]$  和  $d = 1$  可得  $s = \text{bcbc}$ , 最长回文串长度为 3。

**【样例 2】**

见下发文件中的 *caesar/ex\_caesar2.in* 与 *caesar/ex\_caesar2.ans*。

**【样例 3】**

见下发文件中的 *caesar/ex\_caesar3.in* 与 *caesar/ex\_caesar3.ans*。

**【测试点约束】**

本题采用捆绑测试, 只有你通过了一个 subtask 中的所有数据才能获得其分数。

对于全部数据, 满足  $1 \leq |s|, q \leq 10^5$ ,  $1 \leq k \leq 26$ ,  $1 \leq l \leq r \leq |s|$ , 字符串  $s$  仅包含前  $k$  个小写英文字母。

Subtask 1(15%): 满足  $|s|, q \leq 50$ 。

Subtask 2(15%): 满足  $|s|, q \leq 100$ 。

Subtask 3(25%): 满足  $|s|, q \leq 1000$ 。

Subtask 4(15%): 满足  $|s| \leq 5000$ ,  $l = r$ 。

Subtask 5(30%): 无额外限制。