

2024 年天府第七中学信息学竞赛

提高组赛前模拟考试

(请选手务必仔细阅读本页内容)

时间：2024 年 10 月 24 日 14:00–16:00

题目名称	高维字符串匹配	k短路	机器人	集合
题目类型	传统型	传统型	传统型	传统型
目录	dimension	graph	robot	set
可执行文件名	dimension	graph	robot	set
输入文件名	dimension.in	graph.in	robot.in	set.in
输出文件名	dimension.out	graph.out	robot.out	set.out
每个测试点时限	1.0秒	1.0秒	1.0秒	1.0秒
内存限制	512 MB	512 MB	512 MB	512 MB
子任务数目	10	20	20	20
测试点是否等分	是	否	是	是

注意事项与提醒 (请选手务必仔细阅读)

- 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. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
- 8. 每道题目所提交的代码文件大小限制为 **100KB**。
- 9. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
- 10. 不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理。

高维字符串匹配(dimension)

问题描述

在信息科学和计算机视觉中，处理多维数据是一个常见的任务。想象一下，我们正在研究一种新型的医学成像技术，该技术可以生成一个多维字符立方体，表示不同组织的特征。为了分析这种图像，我们需要在这个立方体中识别特定的模式或结构。这类似于在一个高维字符立方体中寻找一个特定子立方体的位置和出现次数。这个问题在其他领域中也有类似应用，比如在材料科学中，通过分析晶体结构来识别特定的原子排列。

现给定两个 k 维字符立方体 A, B ，求 B 在 A 中作为子立方体的出现次数。

立方体 A 的第 i 维大小为 n_i ， A 中的每个包含字符的位置可以用 k 维下标 (i_1, \dots, i_k) 表示（其中 $1 \leq i_j \leq n_j$ ）。

称 A 的一维字符串表示为，一个长为 $\prod_{i=1}^k n_i$ 的字符串

$a_{(1,1,\dots,1,1)}, a_{(1,1,\dots,1,2)}, \dots, a_{(1,1,\dots,1,n_k)}, a_{(1,1,\dots,2,1)}, a_{(1,1,\dots,2,2)}, \dots, a_{(n_1,n_2,\dots,n_k)}$ 。

换句话说， A 的一维字符串表示是将所有下标按字典序排序后，将每个下标对应的字符依次相连，得到的字符串。

立方体 B 的第 i 维大小为 m_i ，其余定义和 A 相似。

输入格式

从文件 `dimension.in` 中读入数据。

第一行一个正整数 k ，描述 A 和 B 的维数。

第二行 k 个正整数 n_1, \dots, n_k ，第 i 个正整数 n_i 表示字符立方体 A 的第 i 维大小。

第三行一个长为 $\prod_{i=1}^k n_i$ 的字符串，表示 A 的一维字符串表示。

第四行 k 个正整数 m_1, \dots, m_k ，第 i 个正整数 m_i 表示字符立方体 B 的第 i 维大小。

第五行一个长为 $\prod_{i=1}^k m_i$ 的字符串，表示 B 的一维字符串表示。

输出格式

输出到文件 `dimension.out` 中。

一行一个正整数，表示答案。

样例输入1

```
1
7
abababa
3
aba
```

样例输出1

```
3
```

样例1解释

当 $k = 1$ 时，原问题为字符串匹配问题。

字符串 `aba` 在字符串 `abababa` 的三个子串 $[1, 3]$, $[3, 5]$, $[5, 7]$ 处出现。

样例2输入

```
3
2 2 2
aaaabbbb
1 1 1
b
```

样例2输出

```
2
```

样例 2 解释

A 为字符串矩阵：

```
aaba
abaa
baaa
```

B 为字符串矩阵：

```
ab
ba
```

B 在 A 的两个子矩阵（以 $(1, 2)$ 为左上角、 $(2, 3)$ 为右下角的子矩阵 和 以 $(2, 1)$ 为左上角、 $(3, 2)$ 为右下角的子矩阵）处出现。

样例3输入

```
3
2 2 2
aaaabbbb
1 1 1
b
```

样例3输出

```
4
```

样例 3 解释

A 为字符串立方体，其第一个面为字符串矩阵：

```
aa
aa
```

其第二个面为字符串矩阵：

bb
bb

提示

对于下标 $i = (i_1, \dots, i_k)$, 称 A 在下标 i 处于高维字符矩阵 B 匹配, 当且仅当:

- 对于所有 $1 \leq p \leq k$, 有 $1 \leq i_p \leq n_p - m_p + 1$;
- 对于所有 $j = (j_1, \dots, j_k)$, 其中 $1 \leq j_p \leq m_p$ ($1 \leq p \leq k$), 都有
$$A_{(i_1+j_1-1, \dots, i_k+j_k-1)} = B_{(j_1, \dots, j_k)}.$$

对于所有数据: $1 \leq m_i \leq n_i \leq 8$, $1 \leq k \leq 5$, 字符集为 `ab`。

对于30%的数据, $k = 1$ 。

对于另外30%的数据, $k = 2$ 。

对于100%的数据, $k \leq 5$ 。

k短路(graph)

问题描述

在现代都市中，交通网络的规划与管理是城市运营的重要一环。假设你是一个城市交通规划师，负责设计一座大型城市的紧急救援系统。这个系统需要确保从医院（起点 S ）到任何一个可能的事故现场（终点 T ）都有多条可行的路径，以便在紧急情况下迅速应对。

城市的交通网络可以抽象为一个带边权的有向图，其中每个节点代表一个重要的交通枢纽或交叉路口，而每条边则代表连接这些节点的道路，边权通常反映通行时间、路程或交通流量。这种抽象模型帮助规划师更好地理解交通流动，并优化路线选择。

为了避免复杂的交通状况和潜在的交通瓶颈，城市规划要求，每个交叉路口至多存在于一个简单环内，从而减少不必要的循环。与此同时，任意两个交叉路口之间至多存在一条简单路径，这样的设计可以简化导航系统，减轻司机的负担。

在这样一个背景下，你需要开发出从医院到事故现场的多条备选路径。特别是在道路被阻塞或发生自然灾害后，找到次优的路径（即第 k 短路径）可以保证救援工作的顺利进行。这不仅能提高紧急响应的效率，还能为城市的日常交通提供有力的支持。

此外，考虑到城市规模和交通复杂度的增加，系统还需要具备扩展性，能够处理较大规模的图和需求。因此，我们需要一个算法来求出从 S 到 T 的第 k 短路径的长度，或者判断是否存在这样的一条路径。通过这样的设计，我们希望在未来的城市交通中，这样的系统能够在危机时刻提供多一重保障。

输入格式

从文件 `graph.in` 中读入数据。

第一行三个整数 n, m, k ，其中 n, m 分别表示图的点数和边数。

第二行两个不同的正整数 S, T 。

接下来 m 行，一行三个正整数 u, v, w ，表示 u 到 v 有一条边权为 w 的有向边。

输出格式

输出到文件 `graph.out` 中。

一行一个整数，表示答案。如果第 k 短路径不存在，则输出 -1 。

样例输入1

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

样例输出1

```
3
```

样例输入2

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

样例输出2

```
15
```

样例 2 解释

从 1 到 3 的第 3 短的路径是 $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 3$ 。

样例输入3

```
2 1 2
1 2
1 2 1
```

样例输出3

```
-1
```

提示

本题开启子任务捆绑测试。

对于所有数据： $2 \leq n \leq 50$ ， $0 \leq m \leq n(n-1)$ ， $1 \leq k \leq 10^{12}$ ， $S \neq T$ ，边权是 $[1, 9]$ 间的整数。

对于subtask1 (25pts)：存在 $n \leq 5, k \leq 10$

对于subtask2 (15pts)：至多存在一个可以被 S 到达，且可以到达 T 的简单环。

对于subtask3 (30pts)：至多存在两个可以被 S 到达，且可以到达 T 的简单环。

对于subtask4 (30pts)：至少存在三个可以被 S 到达，且可以到达 T 的简单环。

机器人(robot)

问题描述

假设你是一个机器人编程大赛的参赛选手，你的任务是编写一个程序来控制一个机器人在一个网格上的移动。这个网格是一个 $n \times n$ 的方格，机器人从左下角的起点 $(0, 0)$ 出发，目标是到达右上角的终点 (n, n) 。机器人每次只能向右或向上移动一个单位距离，模拟了在一个二维空间中的基本移动操作。

这个比赛设定了一些特别的规则来增加挑战性：

- 路径约束：**机器人不能移动到对角线 $y = x$ 的上方，但可以接触到对角线。这意味着机器人在整个移动过程中，始终需要在或者在对角线的下方。
- 方向一致性：**为了模拟真实世界中的某些机械限制或任务需求，比赛还要求机器人在某些特定步骤中必须重复之前某一步的移动方向。例如，第 i 步的移动方向必须与之前的第 p_i 步的移动方向一致（其中 $p_i \leq i$ ）。这可能是为了模拟机器人在特定任务中的重复操作，或者是因为机械臂的结构限制。

你需要编写一个程序来计算从起点到终点的所有合法路径的数量。每条合法路径都必须满足上述两个约束条件。

输入格式

从文件 `robot.in` 中读入数据。

第一行一个正整数 n 。

第二行 $2n$ 个正整数，第 i 个数表示 p_i 。特别地，若 $p_i = i$ ，可以认为第 i 步的移动方向没有额外限制。

输出格式

输出到文件 `robot.out` 中。

一行一个正整数，表示答案。

样例输入1

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

样例输出1

```
2
```

样例 1 解释

用长为 $2n$ 的字符串表示一条路径，其中第 i 位为 R 表示第 i 步向右走，为 U 表示第 i 步向上走。

两条合法路径分别是：RRURUU，RURURU。

样例输入2

```
1
1 1
```

样例输出2

```
0
```

提示

对于20%的数据， $n \leq 10$ 。

对于另外30%的数据， $n \leq 18$ 。

对于另外20%的数据， $n \leq 23$ 。

对于15%的数据，保证所有 $p_i = i$ 。

对于100%的数据， $1 \leq n \leq 25$ 。

集合 (set)

问题描述

你的教练最近又出差了，临行前委托你负责统计你机房同学本次 OI 考试的成绩。为了更好地分析大家的成绩分布和发现问题，以便针对性地提升，你决定将同学们的成绩进行分段统计。

考虑到同学们的成绩种类繁多，你采用了一种特殊的分段统计方法：

1. 累加求和：假设有 n 个同学的成绩序列 $a = \{a_1, a_2, \dots, a_n\}$ ，你需要构造一个新序列 $b = \{b_0, b_1, \dots, b_n\}$ ，其中 $b_i = \lfloor \frac{\sum_{j=1}^i a_j}{10} \rfloor$ 。这样，序列 b 的每个元素就代表一个分数区间。
2. 区间划分：你希望统计出这些区间的不同编号个数（即不同的 b_i 值的个数）。

为了获得最优的统计结果，你可以调整成绩序列 a 的顺序，使得序列 b 中不同编号的个数最多。

此外，你还需要考虑不同的成绩组合进行统计。也就是说，你需要统计所有成绩的子集，并计算每个本质不同的子集的最优排列下的区间编号数的最大值。最后，将所有这些值求和，并对结果取模 998244353。

形式化来说：给定可重集合 A ，对其所有本质不同的子集 B ，求 $f(B)$ 之和对 998244353 取模后的值。

(称两个可重集本质不同，当且仅当存在一种元素，其在两个可重集中的出现次数不同)

输入格式

从文件 `set.in` 中读入数据。

第一行一个正整数 n ，表示可重集 A 的大小。

第二行 n 个正整数，描述了 A 中的每个元素。记第 i 个数为 a_i ，则 $A = \{a\}$ 。

输出格式

从文件 `set.out` 中读入数据。

一行一个正整数，表示答案。

样例输入1

```
1
11
```

样例输出1

```
3
```

样例 1 解释

当 $B = \emptyset$ 时，序列 a 长为 0，对应的序列 $b = \{0\}$ ，故 $f(B) = 1$ ；

当 $B = \{11\}$ 时，序列 a 唯一，对应的序列 $b = \{0, 1\}$ ，故 $f(B) = 2$ 。

样例输入2

```
2
9 11
```

样例输出2

```
7
```

样例 2 解释

$f(\emptyset) = 1$, $f(\{9\}) = 1$, $f(\{11\}) = 2$ 。

当 $B = \{9, 11\}$ 时，令 $a = \{11, 9\}$ ，则 $b = \{0, 1, 2\}$ ， $f(B)$ 取到最大值 3。

样例输入3

```
11
1 2 3 4 5 6 7 8 9 10 11
```

样例输出3

```
7886
```

提示

注意特殊的值域

对于20%的数据， $n \leq 40$ 。

对于另外10%的数据， $a_i \leq 10$ 。

对于另外40%的数据， $n \leq 100$ 。

对于100%的数据， $1 \leq n \leq 2000$, $1 \leq a_i \leq 11$ 。