

图最短路问题 (warm-up)

BFS最短路

时/空限制：1s/64MB

有无测评机：有

给定一个 n 个点 m 条边的有向图，图中可能存在重边和自环。

所有边的长度(权重)都是1，点的编号为 $1 \sim n$ 。

请你求出1号点到 n 号点的最短距离，如果从1号点无法走到 n 号点，输出 -1 。

输入格式

第一行包含两个整数 n 和 m 。

接下来 m 行，每行包含两个整数 a 和 b ，表示存在一条从 a 走到 b 的长度为1的边。

输出格式

输出一个整数，表示1号点到 n 号点的最短距离。

数据范围

$$1 \leq n, m \leq 10^5$$

输入样例：

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

输出样例：

```
1
```

朴素版Dijkstra

时/空限制：1s/64MB

有无测评机：有

给定一个 n 个点 m 条边的有向图，图中可能存在重边和自环，所有边权均为正值。

请你求出1号点到 n 号点的最短距离，如果无法从1号点走到 n 号点，则输出 -1 。

输入格式

第一行包含整数 n 和 m 。

接下来 m 行每行包含三个整数 x, y, z ，表示存在一条从点 x 到点 y 的有向边，边长为 z 。

输出格式

输出一个整数，表示1号点到 n 号点的最短距离。

如果路径不存在，则输出 -1 。

数据范围

$$1 \leq n \leq 500$$

$$1 \leq m \leq 10^5$$

图中涉及边长均不超过10000。

输入样例：

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

输出样例：

```
3
```

堆优化版Dijkstra

时/空限制：1s/64MB

有无测评机：有

给定一个 n 个点 m 条边的有向图，图中可能存在重边和自环，所有边权均为正值。

请你求出1号点到 n 号点的最短距离，如果无法从1号点走到 n 号点，则输出 -1 。

输入格式

第一行包含整数 n 和 m 。

接下来 m 行每行包含三个整数 x, y, z ，表示存在一条从点 x 到点 y 的有向边，边长为 z 。

输出格式

输出一个整数，表示1号点到 n 号点的最短距离。

如果路径不存在，则输出 -1 。

数据范围

$1 \leq n, m \leq 1.5 \times 10^5$
图中涉及边长均不小于0，且不超过10000。
数据保证：如果最短路存在，则最短路的长度不超过 10^9 。

输入样例：

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

输出样例：

```
3
```

八数码问题

八数码解存在性问题（DFS）

时/空限制：1s/64MB

有无测评机：无

在一个 3×3 的网格中， $1 \sim 8$ 这8个数字和一个 **x** 恰好不重不漏地分布在这 3×3 的网格中。

例如：

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

在游戏过程中，可以把 **x** 与其上、下、左、右四个方向之一的数字交换（如果存在）。

我们的目的是通过交换，使得网格变为如下排列（称为正确排列）：

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

例如，示例中图形就可以通过让 **x** 先后与右、下、右三个方向的数字交换成功得到正确排列。

交换过程如下：

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

现在，给你一个初始网格，请你求出得到正确排列至少需要进行多少次交换。

输入格式

输入占一行，将 3×3 的初始网格描绘出来。

例如，如果初始网格如下所示：

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

则输入为： 1 2 3 x 4 6 7 5 8

输出格式

输出该八数码问题是否有解，有解： 1；无解： 0

输入样例：

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

输出样例

```
1
```

八数码问题（BFS，Dijkstra）

时/空限制：1s/64MB

有无测评机：有

在一个 3×3 的网格中，1 ~ 8这8个数字和一个 x 恰好不重不漏地分布在这 3×3 的网格中。

例如：

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

在游戏过程中，可以把 x 与其上、下、左、右四个方向之一的数字交换（如果存在）。

我们的目的是通过交换，使得网格变为如下排列（称为正确排列）：

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

例如，示例中图形就可以通过让 `x` 先后与右、下、右三个方向的数字交换成功得到正确排列。

交换过程如下：

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

现在，给你一个初始网格，请你求出得到正确排列至少需要进行多少次交换。

输入格式

输入占一行，将 3×3 的初始网格描绘出来。

例如，如果初始网格如下所示：

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

则输入为： `1 2 3 x 4 6 7 5 8`

输出格式

输出占一行，包含一个整数，表示最少交换次数。

如果不存在解决方案，则输出 -1 。

输入样例：

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

输出样例

```
19
```

八数码问题（A star）

时/空限制：1s/64MB

有无测评机：有

在一个 3×3 的网格中， $1 \sim 8$ 这8个数字和一个 `x` 恰好不重不漏地分布在这 3×3 的网格中。

例如：

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

在游戏过程中，可以把 `x` 与其上、下、左、右四个方向之一的数字交换（如果存在）。

我们的目的是通过交换，使得网格变为如下排列（称为正确排列）：

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

例如，示例中图形就可以通过让 `x` 先后与右、下、右三个方向的数字交换成功得到正确排列。

交换过程如下：

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

把 `x` 与上下左右方向数字交换的行动记录为 `u`、`d`、`l`、`r`。

现在，给你一个初始网格，请你通过最少的移动次数，得到正确排列。

输入格式

输入占一行，将 3×3 的初始网格描绘出来。

例如，如果初始网格如下所示：

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

则输入为： `1 2 3 x 4 6 7 5 8`

输出格式

输出占一行，包含一个字符串，表示得到正确排列的完整行动记录。

如果答案不唯一，输出任意一种合法方案即可。

如果不存在解决方案，则输出 `unsolvable`。

输入样例：

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

输出样例

ullddrurdllurdruldr