

**Q11. Path Finder (60 marks)**

Consider that a robot operates within a 2-dimensional  $200 \times 200$  grid. Each grid cell, say grid cell  $i$ , is labelled with a coordinate  $(x_i, y_i)$ , where  $1 \leq x_i, y_i \leq 200$ .

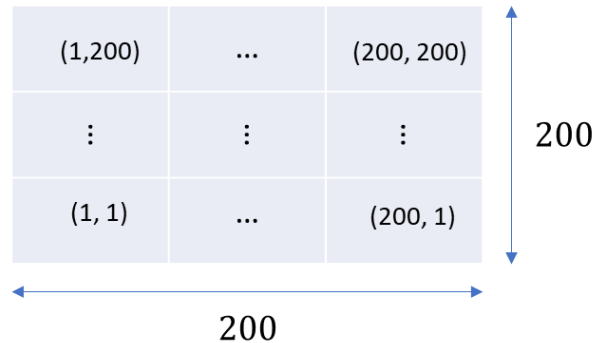


Figure Q11: Grid Environment

The robot will commence its journey at a start point, denoted as  $(x_0, y_0)$ , and it is aimed to travel to reach a specified target point,  $(x_t, y_t)$ . Note that both the start point and target point are guaranteed to be within the grid.

However, the robot cannot move arbitrarily, but only execute a series of movements encoded in a set,  $\mathbf{A}$ . That is, each element of  $\mathbf{A}$  is a pair  $(\Delta x, \Delta y)$  that indicates a valid movement of  $\Delta x$  steps in the  $x$ -direction and  $\Delta y$  steps in the  $y$ -direction, where  $-200 \leq \Delta x, \Delta y \leq 200$ .

Suppose that **a valid move** is defined as an execution of one of the movements given in  $\mathbf{A}$ , and the robot cannot move outside the grid boundaries. The task is to determine the minimum number of moves the robot is required to make in order to arrive at  $(x_t, y_t)$  from  $(x_0, y_0)$ . Note also each movement in set  $\mathbf{A}$  can be repeated multiple times if necessary.

If it is impossible for the robot to reach the target point using the movements in set  $\mathbf{A}$ , your program should return -1.

**For example:**

- **Given that**

- Initial position:  $(x_0, y_0) = (3, 3)$
- Target position:  $(x_t, y_t) = (5, 5)$
- Number of distinct movements:  $n = 2$
- Movements in set  $\mathbf{A}$ :
  - Move right by one step:  $(\Delta x, \Delta y) = (1, 0)$
  - Move up by one step:  $(\Delta x, \Delta y) = (0, 1)$

- **Solution:**

- The minimum number of moves the robot needs to take to reach the target positions is 4. That is, move two steps with  $(0, 1)$  and two steps with  $(1, 0)$ . The order of the moves does not matter in this case.

**Write a programme to****Input, in sequence:**

- (1) Initial position  $x_0$  and  $y_0$ , where  $1 \leq x_0, y_0 \leq 200$ , and both are integers.
- (2) Target position  $x_t$  and  $y_t$ , where  $1 \leq x_t, y_t \leq 200$ , and both are integers.
- (3) An integer  $n$ , denoting the number of distinct movements the robot can perform ( $1 \leq n \leq 10$ ).
- (4)  $n$  pairs of integers, each pair denoting a possible movement  $(\Delta x, \Delta y)$  in set  $A$ .

**Output:**

The minimum number of moves the robot needs to take in order to reach the target position by executing any sequence of the movements in set  $A$  and repeating any movement if necessary. If it's impossible for the robot to reach the target position using the movements in set  $A$ , output -1.

**试题 11. 路径搜寻器 (60 分)**

考虑一个机器人在一个二维的  $200 \times 200$  网格中操作。每个网格单元，例如网格单元  $i$ ，都带有一个坐标  $(x_i, y_i)$ ，其中  $1 \leq x_i, y_i \leq 200$ 。

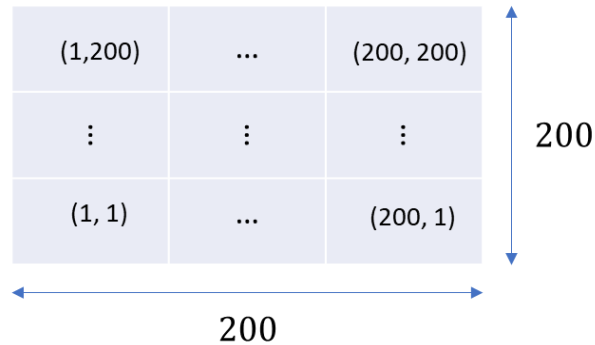


图 Q11: 网格环境

机器人将从一个起点开始它的旅程，且此起点表示为  $(x_0, y_0)$ ，而它的目标是前往一个指定的目标点， $(x_t, y_t)$ 。注意，起始点和目标点都保证在网格内。

然而，机器人不能任意移动，而只能执行一系列在集合  $A$  中编好的动作。也就是说， $A$  的每个元素都是一个移动指令  $(\Delta x, \Delta y)$ ，表示了当执行此移动指令时，机器人会在  $x$  方向上移动  $\Delta x$  步，以及在  $y$  方向上移动  $\Delta y$  步，其中  $-200 \leq \Delta x, \Delta y \leq 200$ 。

假设一个有效的移动动作被定义为执行集合  $A$  中其中一个移动指令，并且机器人不能移动到网格边界之外。您的任务是，通过使用集合  $A$  中的移动指令，寻找机器人从其初始位置到达指定的目标点的最少可能的动作数目。值得注意的是，如有需要，任何一个移动指令可以被重复多次使用。

如果机器人无法使用集合  $A$  中的移动指令到达目标点，则您的程序应返回 -1。

**例子:**

- 倘若:

- 初始位置:  $(x_0, y_0) = (3, 3)$
- 目标位置:  $(x_t, y_t) = (5, 5)$
- 移动指令的数量:  $n = 2$
- 集合  $A$  中的移动指令:
  - 向右移动一步，即  $(\Delta x, \Delta y) = (1, 0)$
  - 向上移动一步，即  $(\Delta x, \Delta y) = (0, 1)$

- 解答:

- 机器人到达目标位置所需的最少动作次数是 4 次。也就是说，用  $(0, 1)$  指令移动两次，再用  $(1, 0)$  指令移动两次。动作的顺序在此例中并不重要。

**试写一程式以****依序输入：**

- (1) 初始位置 $(x_0, y_0)$ ，其中  $1 \leq x_0, y_0 \leq 200$ ，并且它们都是整数。
- (2) 目标位置 $(x_t, y_t)$ ，其中  $1 \leq x_t, y_t \leq 200$ ，并且它们都是整数。
- (3) 一个整数  $n$ ，即机器人可以执行的不同移动指令的数量 ( $1 \leq n \leq 10$ )。
- (4)  $n$  对整数，每对表示集合  $A$  中可能的一个移动指令  $(\Delta x, \Delta y)$ 。

**输出：**

机器人抵达目的所需的最少动作次数。机器人必须利用集合  $A$  中的移动指令来做移动的动作，且若有需要，某个移动指令可重复多次使用。

如果机器人无法使用集合  $A$  中的移动指令以到达目标位置，则输出 -1。

**Examples (例子)**

Input (输入)	Output (输出)
1 1 6 6 3 1 1 2 1 2 2	3
10 10 1 1 2 1 0 0 1	-1
1 1 1 1 1 0 0	0
3 3 5 5 2 1 0 0 1	4
5 5 10 10 1 -1 -1	-1

**Test Cases**

Input (输入)	Output (输出)
1 1 6 6 3 1 1 2 1 2 2	3
10 10 1 1 2 1 0 0 1	-1
1 1 1 1 1 0 0	0
3 3 5 5 2 1 0 0 1	4
5 5 10 10 1 -1 -1	-1
1 1 200 200 1 1 1	199
1 1 200 200 2 1 0 0 1	398
1 1 101 101 4 1 1 -50 0 0 -50 150 150	3

1 1 32 4 10 0 4 4 0 -2 2 0 8 8 0 14 -2 3 5 9 3 -2 14 50 -2	-1
199 199 200 200 1 -1 -1	-1
199 199 200 200 1 1 1	1
1 1 200 200 3 200 200 -1 -1 1 1	199