

# Introdução à Programação

## Aula Prática 3

**Licenciatura em Engenharia Informática FCT/UNL**

**<http://ctp.di.fct.unl.pt/lei/ip/>**

**2021/2022**

# Aula Prática #3

## Problema Mooshak D Robots em Manhattan

# Robots em Manhattan

## Descrição

- Num projeto conjunto internacional, o Departamento de Engenharia Eletrotécnica e de Computadores (DEEC) e o Departamento de Informática (DI) da FCT estão a desenvolver robots capazes de navegar pelas ruas de Manhattan.
- Como as ruas de Manhattan estão dispostas numa grelha, é comum a navegação ser feita através das interseções das ruas (horizontais e verticais). Os robots irão seguir esse modelo de navegação.
- Assim sendo, numa dada interseção de 2 ruas, um robot pode deslocar-se para **Norte**, **Sul**, **Este** ou **Oeste**, uma certa distância **D**, até à próxima interseção.
- Um robot sabe sempre a sua posição na cidade e também sabe qual a distância total percorrida.

# Robots em Manhattan

## Descrição

- Os robots estão equipados com um sistema de marcação de **pontos de interesse**, que armazena as coordenadas de uma localização (um cruzamento de duas ruas) e permite em qualquer instante saber a distância, de acordo com a forma de navegar em Manhattan, da posição atual do robot ao último ponto de interesse armazenado.
- A distância entre duas localizações, usando o modelo de navegação em grelha descrito, também conhecida como a **distância de Manhattan** (ou  $d_1$ ), é a soma dos valores absolutos das diferenças entre as suas coordenadas (x e y):

$$d_1(p,q) = |p_x - q_x| + |p_y - q_y|$$

# Robots em Manhattan

## Descrição

- A distância entre duas localizações, usando o modelo de navegação em grelha descrito, também conhecida como a **distância de Manhattan** (ou  $d_1$ ), é a soma dos valores absolutos das diferenças entre as suas coordenadas (x e y):

$$d_1(p,q) = |p_x - q_x| + |p_y - q_y|$$

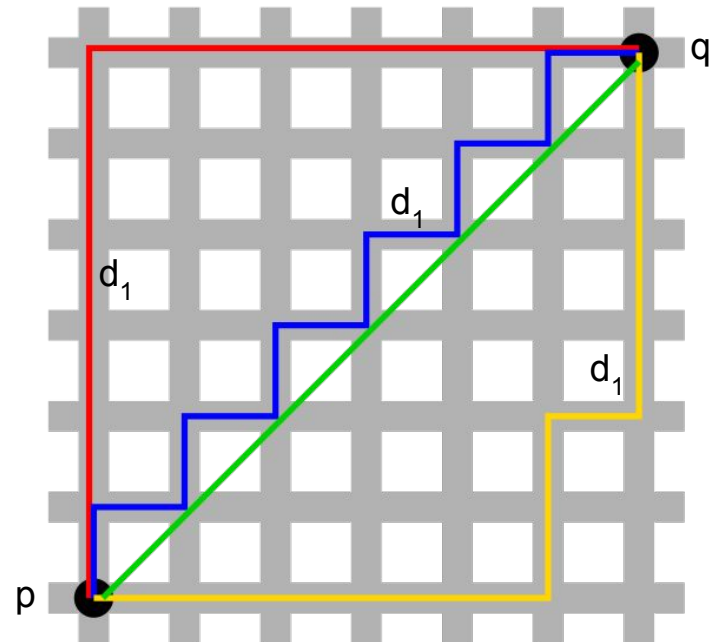


Image by Bogdan Giuscă. Licensed under the [Creative Commons Attribution-Share Alike 3.0 Unported](https://creativecommons.org/licenses/by-sa/3.0/) license.

# Robots em Manhattan

## **Problema:**

- Implementar uma aplicação de controlo de dois robots em Manhattan.

## **Input:**

- O input começa com duas linhas, cada uma com dois números inteiros positivos,  $x$  e  $y$ , que definem a posição inicial de cada robot. A posição inicial é um ponto de interesse do robot. Inicialmente, os robots estão orientados para Norte.
- Seguem-se 5 linhas, cada uma com um comando. Um comando é constituído por uma palavra em maiúsculas, seguida dos argumentos do comando, separados por um espaço. A execução de um comando pode ou não produzir uma mensagem para o utilizador.
- Assuma que o input está bem formado.

# Robots em Manhattan

## Comandos

- `MV N K` (N é 1 ou 2, K é um valor inteiro positivo)

Move o N-ésimo robot, K unidades na sua direção atual. Não produz qualquer mensagem.

- `MD N D` (N é 1 ou 2, D é um dos caracteres N, S, E, O)

Muda a direção do N-ésimo robot para Norte, Sul, Este ou Oeste. Não produz qualquer mensagem.

- `PI N` (N é 1 ou 2)

Marca a posição atual do N-ésimo robot como um ponto de interesse. Não produz qualquer mensagem.

# Robots em Manhattan

## Comandos

- LP N (N é 1 ou 2)

Imprime uma linha com dois números inteiros,  $P_x$   $P_y$ , separados por um espaço, correspondendo às coordenadas da posição atual do N-ésimo robot.

- LDT N (N é 1 ou 2)

Imprime uma linha com um número inteiro D, a distância total percorrida pelo N-ésimo robot.

- LDPI N (N é 1 ou 2)

Imprime uma linha com um número inteiro D, a distância do N-ésimo robot ao seu último ponto de interesse.



# Robots em Manhattan

## Comandos

- DTMAX

Imprime a mensagem “ROBOT 1” (sem aspas) se o primeiro robot percorreu uma distância total superior à distância total percorrida pelo segundo robot, “ROBOT 2” (sem aspas) se o segundo robot percorreu uma distância total superior, ou “EMPATE” se ambos percorreram a mesma distância.

# Robots em Manhattan

## Exemplo

### Input:

```
0 0
1 0
LP 1
LP 2
LDPI 1
LDPI 2
DTMAX
```

### Output:

```
0 0
1 0
0
0
EMPATE
```

Os robots não se mexeram e o último ponto de interesse de cada robot é a sua posição inicial.

# Robots em Manhattan

## Exemplo

### Input:

```
0 0
10 10
MV 1 10
LP 1
MD 1 E
MV 1 10
LP 1
```

### Output:

```
0 10
10 10
```

O Robot 1 anda inicialmente para Norte (porque é a direção inicial dos robots).

# Robots em Manhattan

## Exemplo

### Input:

```
2 2
2 0
MV 1 7
MD 1 0
MV 1 20
LDPI 1
LDT 1
```

### Output:

```
27
27
```

A distância de Manhattan ao ponto de interesse é 27, que coincide com a distância total percorrida pelo Robot 1.

# Robots em Manhattan

## Exemplo

### Input:

```
0 1
2 3
MV 1 5
MV 2 3
DTMAX
MV 2 2
DTMAX
```

### Output:

```
ROBOT 1
EMPATE
```

Nos primeiros movimentos, os robots deslocam-se, respetivamente, 5 e 3 unidades. Portanto, o Robot 1 percorre uma maior distância. Depois, o Robot 2 anda 2 unidades adicionais e a distância total percorrida de ambos é a mesma.

# Robots em Manhattan

## Exemplos

<b>Input:</b> 0 1 2 3 MV 1 5 DTMAX MV 2 6 DTMAX LDT 1	<b>Input:</b> 100 23 99 1 MD 2 S MV 2 1 PI 2 LDPI 1 LDPI 2	<b>Input:</b> 100 23 99 1 MV 2 1 PI 2 MV 2 10 LDPI 1 LDPI 2
<b>Output:</b> ROBOT 1 ROBOT 2 5	<b>Output:</b> 0 0	<b>Output:</b> 0 10

# Esqueleto da Classe Robot

```
public class Robot {  
    public static final String NORTH = "N";  
    public static final String SOUTH = "S";  
    //...  
  
    private int currentX, currentY;  
    private String currentHeading;  
    private int totalDistance;  
    private int markedX, markedY;  
  
    public Robot(int initX, int initY) {...}  
  
    public int getXPos() {...}  
    public int getYPos() {...}  
    public int getTotalDistance() {...}  
    public int getPIDistance() {...}  
  
    public void move(int distance) {...}  
    public void setHeading(String heading) {...}  
    public void mark() {...}  
}
```

# Robots em Manhattan - Desafio (E)

Duas formas de comando adicionais:

- DR

Imprime uma linha com um número inteiro  $D$ , a distância de Manhattan entre os dois robots.

- LMV

Imprime a mensagem “ROBOT 1” (sem aspas) se o primeiro robot foi o que percorreu a maior distância num único movimento, “ROBOT 2” (sem aspas) se o segundo robot foi o que percorreu a maior distância num único movimento ou “EMPATE” (sem aspas) em caso de empate.



# Robots em Manhattan - Desafio (E)

## Exemplos

### Input:

10 0

0 10

DR

MV 1 5

DR

MV 2 5

LMV

### Output:

20

15

EMPATE

Inicialmente os robots estão a 20 unidades de distância entre si. O Robot 1 desloca-se 5 unidades para Norte, ficando os robots a 15 unidades de distância entre si. O Robot 2 desloca-se 5 unidades para Norte e, portanto, nenhum robot percorreu mais distância que o outro num único movimento.

# Robots em Manhattan - Desafio (E)

## Exemplos

<b>Input:</b> 0 250 0 0 DR MV 2 250 DR MV 1 100 DR	<b>Input:</b> 0 0 0 0 MV 1 5 MV 2 10 MV 1 5 LMV DTMAX
<b>Output:</b> 250 0 100	<b>Output:</b> ROBOT 2 EMPATE

# Mooshak

- O exercício está aberto no Mooshak com a letra D.
- O desafio está aberto no Mooshak com a letra E.
- Ambos encerram às 10h do dia 28 de Outubro.
- Submeta um ZIP (por problema) que contém **apenas** os ficheiros .java que tiver definido para resolver o problema.
- Cada ZIP tem necessariamente de conter um ficheiro Main.java (com o método `main`).