

”

**E-fólio B** | Folha de resolução para E-fólio



**UNIDADE CURRICULAR:** Introdução à Inteligência Artificial

**CÓDIGO:** 21071

**DOCENTE:** José Coelho

**A preencher pelo estudante**

**NOME:** Pedro Pereira Santos

**N.º DE ESTUDANTE:** 2000809

**CURSO:** Licenciatura em Engenharia Informática

**DATA DE ENTREGA:** 23/05/2023

## TRABALHO / RESOLUÇÃO:

Para este problema, temos os requerimentos de aplicar um algoritmo para procura informada. Posto isto, decidi utilizar o algoritmo  $A^*$ , que expande todos os estados possíveis de um estado, passando para o estado, do conjunto de estados total, que tenha menor custo pela função de  $f(\text{custo}) = \text{custo para chegar ao nó } n$  (custo do caminho desde o nó inicial até ao nó  $n$ ) + custo estimado do caminho de menor custo do nó  $n$  ao nó objetivo.

O problema engloba um Mapa  $N \times N$ , onde  $N \in \{5, 7, 9\}$ , de uma cidade, que tem 3 tipos de zonas:

- Zonas Habitacionais.
- Zonas Comuns de Circulação
- Zonas inacessíveis(Não pode haver nem construção nem passagem).

Dentro do programa, estas 3 zonas estão identificadas por 1,0 ou -1, respectivamente.

Temos também as portas das cidades, que ficam situadas nos pontos cardinais N,S,E,O.

Temos o objetivo de fazer com que todas as Zonas Habitacionais tenham acesso a todas as portas das cidades, em que para chegar lá, só se pode passar pelas zonas de circulação. Uma Zona Habitacional fica com acesso a uma porta por avaliação de dois casos:

- Ou está diretamente adjacente a uma na horizontal,vertical ou diagonal.
- Ou consegue passar pelas zonas de circulação, passando uma a uma até chegar a uma porta, onde os movimentos possíveis são em todas as direções, horizontal, vertical e diagonal, em apenas uma zona de cada vez. A verificação se está adjacente a uma porta é igual à de cima.

Para chegar a este objetivo, podemos trocar a posição de uma Zona Habitacional pela posição de uma Zona de Circulação, até atingirmos o objetivo.

Caso seja impossível ter acesso a todas as portas, temos outro critério, minimizar o número de zonas habitacionais que não têm acesso à porta que tem menos acessos.

Para a heurística da procura temos dois custos:

- Custo de movimento de Zona Habitacional por Zona de Circulação, que vale uma unidade.
- Custo por cada Zona Habitacional que não tem acesso à porta com menos acessos, que vale 100 unidades.

Como o custo de movimento das Zonas é unitário e como o número de movimentos poderá ser comparado ao nível de procura, a heurística irá ter um valor de:

$(\text{Zonas Habitacionais sem acesso à porta menos acessível} * 100 + \text{nível})$  unidades

Temos 10 problemas para resolver aplicando o algoritmo. Para guardar os dados das procuras temos uma tabela:

<b>Algoritmo</b> <b>A*</b>	Instância	1	2	3	4	5	6	7	8	9	10
	Avaliações	335	328	1425	-	19633	602	-	-	4687	4909
	Gerações	305	283	985	-	12558	573	-	-	4441	4697
	Custo	3	4	4	-	4	2	-	-	6	4
	Tempo(msec)	38	62	331	-	60000	109	-	-	6876	7557

Melhores Resultados:

Instância 1:

```
A fazer caminho...
10110
11011
10##0
01010
11100
Geracao:305 Expansao:4 Nivel:3 Heuristica:3

10110
11011
00##0
11010
11100
Geracao:192 Expansao:3 Nivel:2 Heuristica:102

10110
11011
01##0
11010
11000
Geracao:2 Expansao:2 Nivel:1 Heuristica:801

01110
11011
01##0
11010
11000
Geracao:1 Expansao:1 Nivel:0 Heuristica:1000
```

## Instância 2:

```
Melhor caminho encontrado.

A fazer caminho...
11101
01#01
0###0
10#01
11011
Geracao:283 Expansao:5 Nivel:4 Heuristica:4

11101
11#01
0###0
10#01
01011
Geracao:227 Expansao:4 Nivel:3 Heuristica:203

11101
11#01
0###0
10#10
01011
Geracao:126 Expansao:3 Nivel:2 Heuristica:602

01101
11#11
0###0
10#10
01011
Geracao:60 Expansao:2 Nivel:1 Heuristica:701

01100
11#11
0###0
11#10
01011
Geracao:1 Expansao:1 Nivel:0 Heuristica:800
```

## Instância 3:

```
Melhor caminho encontrado.

A fazer caminho...
10111
1#001
1###0
001#0
11001
Geracao:985 Expansao:16 Nivel:4 Heuristica:4

10101
1#011
1###0
001#0
11001
Geracao:230 Expansao:4 Nivel:3 Heuristica:503

10100
1#011
1###0
011#0
11001
Geracao:171 Expansao:3 Nivel:2 Heuristica:602

10100
1#011
1###0
011#0
01011
Geracao:2 Expansao:2 Nivel:1 Heuristica:701

01100
1#011
1###0
011#0
01011
Geracao:1 Expansao:1 Nivel:0 Heuristica:700
```

Instância 4:

Esta instância não termina e não consigo resultados com ela. Tentei verificar, mas não conseguir encontrar a razão.

Instância 5:

Esta instância não termina, mas consegue encontrar uma solução de custo 4.

```
Numero de Avaliacoes: 21368  
  
Numero de Gerados: 13799  
  
Custo Final: 4  
  
Tempo de execucao: 60039MS -- 60.039S  
PS C:\Users\Pedro\Desktop\EfolioBIIA>
```

Instância 6:

```
Melhor caminho encontrado.  
  
A fazer caminho...  
1110000  
0101110  
10###01  
01###10  
01###01  
0110101  
0011010  
Geracao:573 Expansao:3 Nivel:2 Heuristica:2  
  
1100000  
0101110  
10###01  
01###10  
11###01  
0110101  
0011010  
Geracao:22 Expansao:2 Nivel:1 Heuristica:101  
  
0100010  
0101110  
10###01  
01###10  
11###01  
0110101  
0011010  
Geracao:1 Expansao:1 Nivel:0 Heuristica:1000  
  
Numero de Avaliacoes: 602  
  
Numero de Gerados: 573  
  
Custo Final: 2  
  
Tempo de execucao: 109MS -- 0.109S  
PS C:\Users\Pedro\Desktop\EfolioBIIA> ./main
```

Instância 7:

Analogamente à Instância 4, esta instância não termina e não consigo resultados com ela. Tentei verificar, mas não conseguir encontrar a razão.

Instância 8:

Analogamente à Instância 4, esta instância não termina e não consigo resultados com ela. Tentei verificar, mas não conseguir encontrar a razão.

Instância 9:

```
Melhor caminho encontrado.

A fazer caminho...
11010011#
10#01101#
11#101#0#
###101#10
011111#10
10#####10
101000#0#
010101#0#
00010000#
Geracao:4441 Expansao:7 Nivel:6 Heuristica:6

11010011#
11#01101#
01#101#0#
###101#10
011111#10
10#####10
101000#0#
010101#0#
00010000#
Geracao:3490 Expansao:6 Nivel:5 Heuristica:305

11110011#
11#01101#
01#101#0#
###101#10
011101#10
10#####10
101000#0#
010101#0#
00010000#
Geracao:2794 Expansao:5 Nivel:4 Heuristica:504
```

```
11110011#
11#01101#
01#101#0#
###101#10
011101#10
10#####10
101000#0#
010101#0#
00010000#
Geracao:2794 Expansao:5 Nivel:4 Heuristica:504
```

```
11110011#
11#11101#
01#101#0#
###101#10
010101#10
10#####10
101000#0#
010101#0#
00010000#
Geracao:2440 Expansao:4 Nivel:3 Heuristica:903
```

```
11110010#
11#11101#
01#101#0#
###101#10
010101#10
10#####10
101000#1#
010101#0#
00010000#
Geracao:1136 Expansao:3 Nivel:2 Heuristica:1902
```

```
11110000#
11#11111#
01#101#0#
###101#10
010101#10
10#####10
101000#1#
010101#0#
00010000#
Geracao:482 Expansao:2 Nivel:1 Heuristica:2401
```

```
01110000#
11#11111#
01#101#0#
###101#10
110101#10
10#####10
101000#1#
010101#0#
00010000#
Geracao:1 Expansao:1 Nivel:0 Heuristica:2800
```

Numero de Avaliaco'es: 4687

Numero de Gerados: 4441

Custo Final: 6

Tempo de execucao: 6876MS -- 6.876S

## Instância 10:

```
Melhor caminho encontrado.

A fazer caminho...
111101111
101011101
110101011
111110111
010011000
101100111
110101011
101101101
111101111
Geracao:4691 Expansao:5 Nivel:4 Heuristica:4

111101111
101011101
110101011
111110111
010010000
101101111
110101011
101101101
111101111
Geracao:3332 Expansao:4 Nivel:3 Heuristica:703

111101111
101011101
110101011
111110111
000010000
111101111
110101011
101101101
111101111
Geracao:1905 Expansao:3 Nivel:2 Heuristica:1402

111101111
101011101
110101011
111101111
000010000
111101111
110101011
101101101
111101111
Geracao:244 Expansao:2 Nivel:1 Heuristica:2101

111101111
101101101
110101011
111101111
000010000
111101111
110101011
101101101
111101111
Geracao:1 Expansao:1 Nivel:0 Heuristica:2800

Numero de Avaliacoes: 4909

Numero de Gerados: 4691

Custo Final: 4

Tempo de execucao: 7210MS -- 7.21S
```



Eu implementei de origem o algoritmo, por isso tenho alguns bugs, que até à data não consegui resolver, entretanto, tenho 6 soluções encontradas com sucesso, uma delas encontra até um custo de 4 mas não melhora e as outras 3, o algoritmo não desenvolve.

Este foi o algoritmo que melhor entendi por isso é que resolvi implementá-lo.