



UNIVERSIDADE DE ÉVORA

INTELIGÊNCIA ARTIFICIAL

---

# Relatório do 1º Trabalho Prático

---

*Autor:*

Pedro NUNES, 31240

Tiago MARTINHO, 35735

*Docente:*

Paulo QUARESMA

Março de 2018

# Índice

1	Pesquisa não Informada	2
2	Pesquisa Informada	4

# 1 Pesquisa não Informada

## Definição do problema

Suponha que o labirinto tem dimensão 4x4 e que o agente está no quarto (1,1) e pretende ir para o quarto (4,4) mas que as portas entre os quartos (1,1) e (1,2), (2,1) e (2,2), (3,1) e (4,1), (3,2) e (3,3), e (4,2) e (4,3) estão bloqueadas. Neste caso as portas bloqueadas foram abordadas como statments, como se fosse uma base de dados.

A			
			O

## Estados:

### Estado Inicial:

O estado inicial é composto por dois argumentos, a dimensão do labirinto e a posição inicial do agente.

```
%estado_inicial(Estado) -> dim, pos_actual
```

```
estado_inicial([dim(4, 4), (1, 1)]).
```

### Estado Final:

O estado final é composto por dois argumentos, mas só a posição final do agente é essencial.

```
%estado_final(Estado)
```

```
estado_final([_, (4, 4)]).
```

## Operadores:

Temos 4 operações possíveis: norte, sul, este, oeste.

```
%representacao dos operadores
```

```
%op(Eact, OP, Eseg, Custo)
```

```
%Norte
```

```
op([dim(Xmax,Ymax), (Xcur, Ycur)], norte, [dim(Xmax,Ymax), (Xcur, Yn)], 1): -  
    Yn is Ycur - 1,  
    Yn > 0,  
    \+bloqueado( (Xcur, Ycur), (Xcur, Yn) ),  
    \+bloqueado( (Xcur, Yn), (Xcur, Ycur) ).
```

```

%Este
op ([ dim(Xmax,Ymax) ,( Xcur , Ycur ) ] , este , [ dim(Xmax,Ymax) ,( Xn , Ycur ) ] , 1): –
    Xn is Xcur + 1 ,
    Xn ≤ Xmax ,
    \+bloqueado( (Xcur , Ycur) , (Xn , Ycur) ) ,
    \+bloqueado( (Xn , Ycur) , (Xcur , Ycur) ) .

%Sul
op ([ dim(Xmax,Ymax) ,( Xcur , Ycur ) ] , sul , [ dim(Xmax,Ymax) ,( Xcur , Yn ) ] , 1): –
    Yn is Ycur + 1 ,
    Yn ≤ Ymax ,
    \+bloqueado( (Xcur , Ycur) , (Xcur , Yn) ) ,
    \+bloqueado( (Xcur , Yn) , (Xcur , Ycur) ) .

%Oeste
op ([ dim(Xmax,Ymax) ,( Xcur , Ycur ) ] , oeste , [ dim(Xmax,Ymax) ,( Xn , Ycur ) ] , 1): –
    Xn is Xcur – 1 ,
    Xn > 0 ,
    \+bloqueado( (Xcur , Ycur) , (Xn , Ycur) ) ,
    \+bloqueado( (Xn , Ycur) , (Xcur , Ycur) ) .

```

### (a) Algoritmo de pesquisa não informada mais eficiente para resolver o problema.

O algoritmo de pesquisa não informada utilizado foi a pesquisa em profundidade sem ciclos, uma vez que os custos das operações são sempre unitários, tratar-se então duma situação óptima para o mesmo. Esta pesquisa é uma pesquisa não informada uma vez que os únicos custos de transição de estados a serem avaliados são os de um estado para o próximo.

```

pesquisa_largura ([ no(E,Pai,Op,C,P) | _ ] , no(E,Pai,Op,C,P) , V , M) :-
    estado_final(E) ,
    write("Total de estados visitados: \t") , write(V) , nl ,
    write("Maximo de estados em memoria: \t") , write(M) , nl .

pesquisa_largura ([E|R] , Sol , V , M):-
    expande(E,Lseg) ,
    insere_fim(Lseg,R,LFinal) ,
    length(LFinal,Mn) ,
    max(M,Mn,Mmn) ,
    Vn is V + 1 ,
    pesquisa_largura(LFinal,Sol,Vn,Mmn).

expande(no(E,Pai,Op,C,P),L):-
    findall(no(En,no(E,Pai,Op,C,P),Opn,Cnn,P1),
        (op(E,Opn,En,Cn), P1 is P+1, Cnn is Cn+C),
        L).

```

**Número total de estados visitados: 203**

**Número máximo de estados simultâneos em memória: 286**

**Caminho realizado:** este este sul oeste sul este este sul

## 2 Pesquisa Informada

**(a) Heurística admissível para estimar o custo de um estado até à solução para este problema.**

A heurística escolhida para calcular(valor estimado), ou seja o custo de um estado até à solução do problema foi a distância de Manhattan. É dada pela distância entre dois pontos numa grelha baseada num caminho estritamente horizontal e/ou vertical, ou seja, ao longo das linhas de grelha. A representação matemática da heurística é:

$$|x' - x| + |y' - y|$$

**(b) Apresente o código em Prolog do algoritmo de pesquisa informada mais eficiente para resolver este problema usando a heurística definida na alinea anterior.**

```
pesquisa_aux([no(E,Pai,Op,C,CH,P)|_],no(E,Pai,Op,C,CH,P),V,M):-
    estado_final(E),
    write("Total de estados visitados: \t"), write(V), nl,
    write("Maximo de estados em memoria: \t"), write(M), nl.
pesquisa_aux([E|R],Sol,V,M):-
    expande(E,Lseg),
    insere_ordenado(Lseg,R,LFinal),
    length(LFinal,Mn),
    max(M,Mn,Mm),
    Vn is V + 1,
    pesquisa_aux(LFinal,Sol,Vn,Mm).

expande(no(E,Pai,Op,C,CH,P),L):-
    findall(no(En,no(E,Pai,Op,C,CH,P),Opn,Cnn,CHn,P1),
        (op(E,Opn,En,Cn),P1 is P+1,Cnn is Cn+C,heur(En,H),CHn is Cnn+H),
        L).
```

**Número total de estados visitados:** 192

**Número máximo de estados simultâneos em memória:** 271

**Caminho realizado:** este este sul oeste sul sul este este