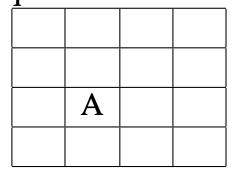
Universidade de Évora Escola de Ciências e Tecnologias

Inteligência Artificial 1º Trabalho

# Resolução de Problemas como problemas de pesquisa no espaço de estados

****

Trabalho realizado por:

- Cláudia Dias nº35308

- João Queimado nº

Ano letivo 2018/2019

# Introdução

Uma pesquisa não informada só usa a informação disponível na definição do problema. Exemplos de pesquisa não informada:

* Pesquisa em largura (Breadth-first search);
* Pesquisa de custo uniforme (Uniform-cost search);
* Pesquisa em Profundidade (Depth-first search);
* Pesquisa em profundidade limitada (Depth-limited search);
* Pesquisa em Profundidade iterativa (Iterative deepening search).

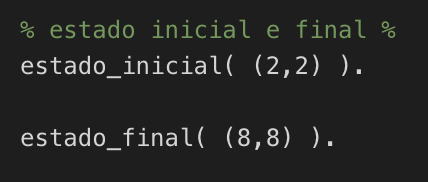
Uma pesquisa informada utiliza conhecimento específico sobre o problema para encontrar soluções de forma mais eficiente. Procura pelo melhor resultado. Exemplos de pesquisa informada:

* Pesquisa ansiosa;
* Best-first search;
* Pesquisa A\*.

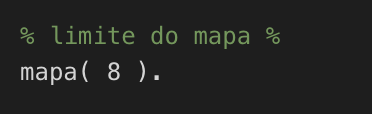
# Resposta às Questões

## Questão 1: Represente em Prolog o espaço de estados e os operadores de transição de estados para este problema:

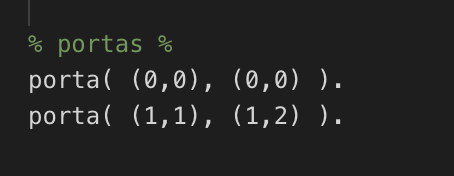
Para representar o espaço de estados utilizámos um tuplo para o estado inicial que abrange as coordenadas da sala de partida, neste caso é (2,2), que representa X e Y respetivamente. Para representar o estado final utilizámos novamente um tuplo que contém as coordenadas da sala de chagada, por exemplo (8,8).



Para definir o tamanho da matriz utilizámos o predicado seguin

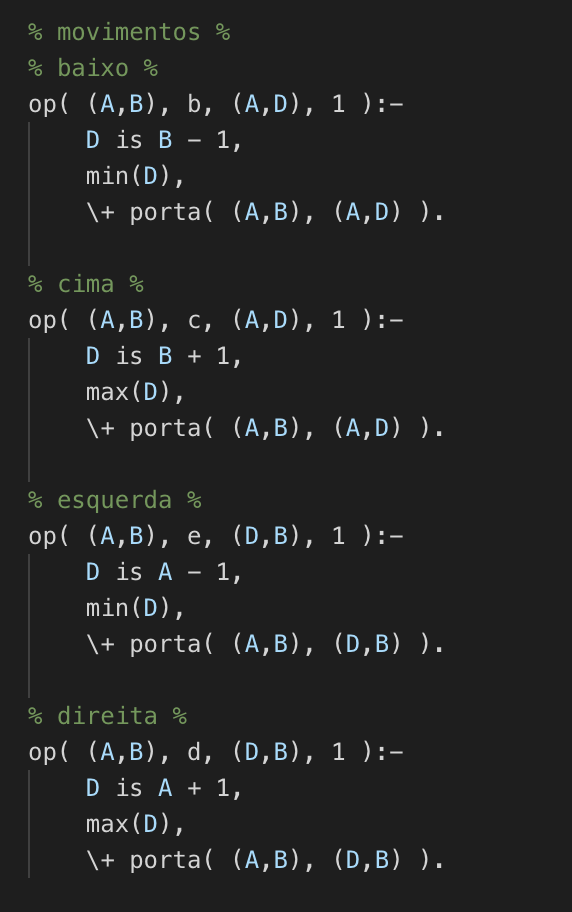


Definimos as transições que não se podem realizar, ou seja, as portas bloqueadas da seguinte maneira:



Temos quatro operações para permitir as transições, sendo elas:

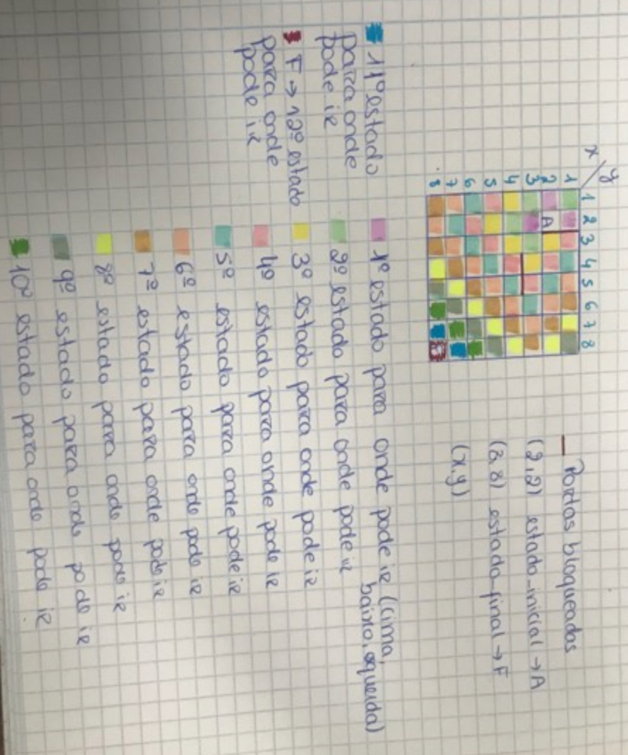
* Baixo
* Cima
* Esquerda
* Direita



### Alínea (a):

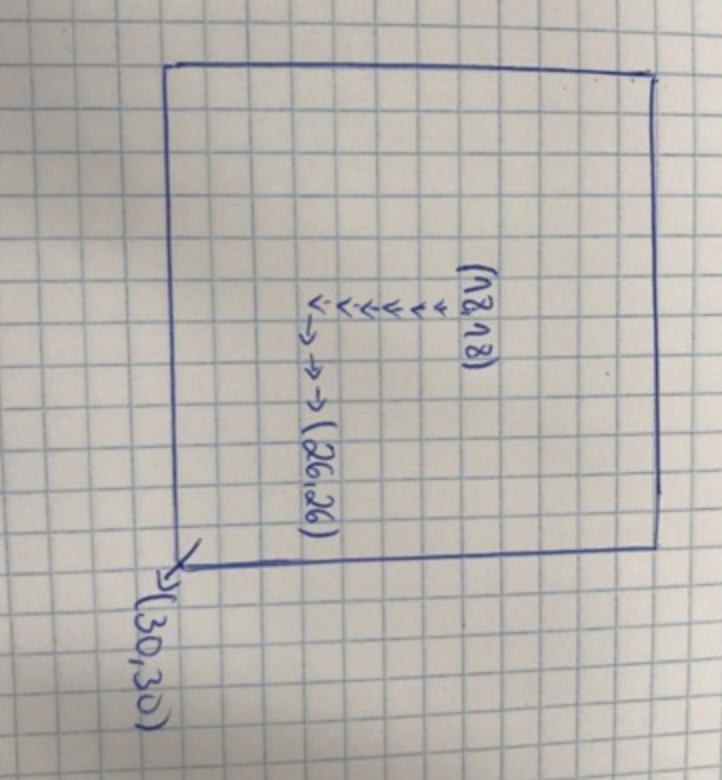
Usando o exercício acima com uma sala 8 por 8, a iniciar na sala (2,2) e terminando na sala (8,8), com as portas bloqueadas entre (1,2) e (1,3), (2,3) e (2,2), (3,4) e (4,4) e (4,5) e (3,5), a aparência seria assim:

O agente irá avaliar os possíveis movimentos para poder chegar ao estado final, não passando dos limites do mapa nem pelas portas bloqueadas, por exemplo, no estado inicial A este pode andar para cima, para baixo ou para a esquerda não podendo ir para a direita porque a porta está bloqueada.



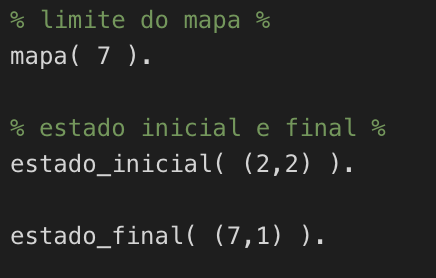
O algoritmo mais adequado para resolver este problema é a pesquisa em profundidade iterativa, porque percorre a árvore desde o primeiro nó até ao nó de chegada mais rapidamente.

Por exemplo do nó (18,18) para o nó (26,26) o algoritmo vai descendo nó a nó e depois percorre os nós para a direita até chegar ao estado final (26,26). Como por exemplo a figura abaixo é uma das soluções.

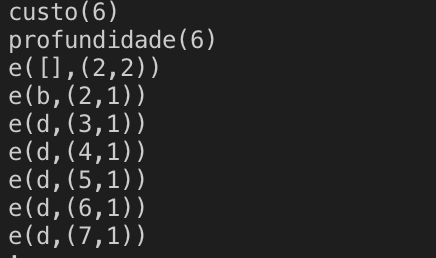


### Alínea (b):

Como o nosso trabalho está a ter alguns problemas só conseguimos obter resultados com:

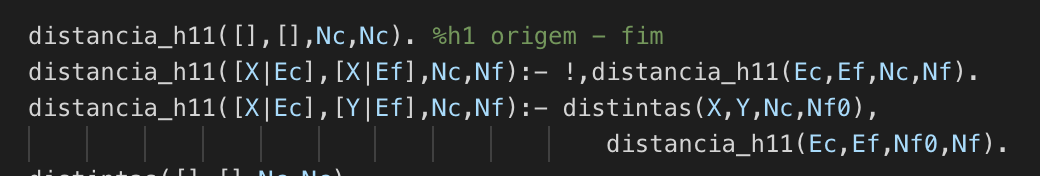
****

O resultado obtido foi:

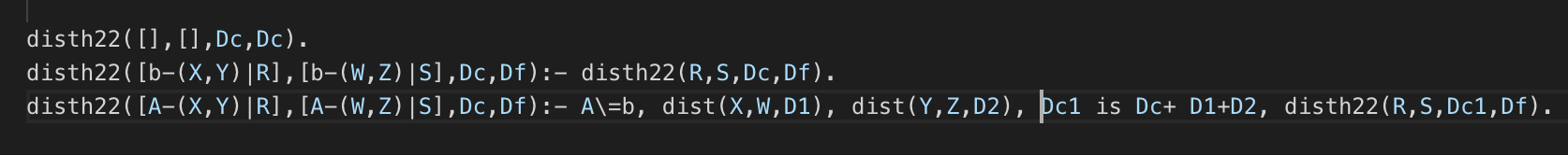


**Questão 2**

A heurística 1 que começa na origem e percorre todos os nós até ao nó final.



A heurística 2 é a heurística de Manhattan que é a soma da distância de Manhattan (isto é o número de quadrados até à localização desejada para cada peça).



## Questão 3

Não conseguimos resolver esta questão.