

Resolução da Ficha 5

- 1) É um problema de estado único.
- 2) Formulando o problema:
 - a. Conjunto de Estados: conjunto de pares ordenados de inteiros (X,Y) , em que X representa o número de litros de água no jarro de 8 litros de capacidade e Y representa o número de litros de água no jarro de 5 litros de capacidade.
 - b. Estados iniciais: jarros vazios
 - c. Estados finais: jarro de 8 litros com apenas 4 litros, jarro de 5 litros com apenas 4 litros
 - d. Operadores:
 - esvaziar um balde
 - encher (completamente) um balde
 - despejar um balde para o outro até que o segundo fique cheio
 - despejar um balde para o outro até que o primeiro fique vazio
 - e. Transições (exemplo):

# transição	Estado inicial	Condição	Estado Final	Descrição
1	(0,0)	If $(x < 8)$	(8,y)	Encher X
2	(0,0)	If $(y < 5)$	(x,5)	Encher Y
3	(8,0)	If $(x > 0)$	(0,0)	Esvaziar X
4	(8,0)	If $(y < 5)$	(8,5)	Encher y
5	(8,0)	If $(x > 0 \text{ e } y = 0)$	(3,5)	Despejar parte da água do jarro de 8 litros para o de 5 litros
6	(0,5)		

- f. Teste Objetivo: Um dos dois jarros deve conter somente 4 litros de água
- g. Custo: Cada ação custa 1 unidade

- 3) Pesquisa em Largura:

- a. Os nós, em cada nível da árvore, são completamente examinados antes de se mover para o próximo nível
- b. Uma busca em largura encontrará sempre o caminho menor entre o estado inicial e o estado-objectivo (final)
- c. O menor caminho é o caminho com o menor número de passos (não confundir com o caminho de menor custo).
- d. É necessário manter um conjunto de nós candidatos alternativos e não apenas o nó candidato como na busca em profundidade
- e. O conjunto de nós candidatos é constituído por todos o nós de um dado nível da árvore de busca
- f. Entretanto, mesmo este conjunto de caminhos candidatos não é suficiente então deve-se extrair um caminho de solução do processo de busca.
- g. Portanto, ao invés de manter-se um conjunto de nós candidatos, deve-se manter um conjunto de caminhos-candidatos.

Exemplo:

1. Conjunto de Candidatos Inicial:
[[(0,0)]]
2. Gera extensões de [(0,0)]:
[[(8,0),(0,0)], [(0,5), (0,0)]] (1.º nível da árvore)
3. Remove o 1.º caminho candidato [(8,0),(0,0)], e gera extensões deste caminho, colocando-as no final do conjunto:

[[(0,5), (0,0)], [(0,0),(8,0),(0,0)], [(3,5),(8,0),(0,0)], [(8,5),(8,0),(0,0)]] (2.º nível da árvore)
4. Remove [(0,5), (0,0)] e acrescenta a sua extensão ao final do conjunto de candidatos, produzindo:

[[[(0,0), (8,0),(0,0)], [(3,5),(8,0),(0,0)], [(8,5),(8,0),(0,0)], [(0,0),(0,5),(0,0)], [(5,0),(0,5),(0,0)], [(8,5),(0,5),(0,0)]]] (2.º nível da árvore)
5. E continua a repetir o processo de extensão até a busca encontrar um caminho que tenha o nó-objectivo/final à cabeça ([_,4] ou (4,_)) e assim este caminho apresentado com uma solução.