Desafios Inteligência Artificial

Aluno: Guilherme Boroni Pereira

Desafio 1:

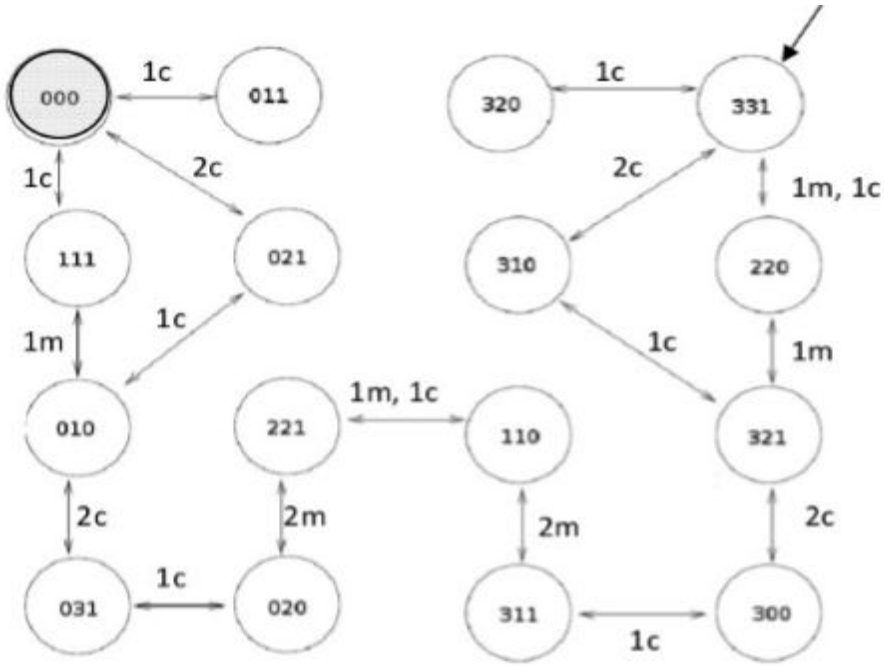
a)

- Estado inicial: Três missionários e três canibais estão em um lado do rio.
- Teste de objetivo: Três missionários e três canibais estão do outro lado do mesmo rio.
- Função sucessor: Os sucessores de um estado são todos os estados aonde se chega após realizar a ação de mover uma ou duas pessoas (canibais ou missionários) e um barco de um lado para o outro do rio, sem que o número de missionários de um lado fique menor do que o número de canibais.
- Função de custo: Número total de ações. Cada movimento de um lado até o outro lado do rio tem custo 1.

b)

Espaço de estado O estado inicial e a função sucessor nos permitem definir o espaço de estados (todas as possíveis configurações válidas de canibais e missionários em cada lado do rio) = 16 possíveis estados

**Explicação da numeração : 321 = 3 - Missionários, 2 -Canibais, 1 - Barco na posição inicial



c) Busca em Profundidade, visto que a árvore/grafo observado a partir do diagrama do espaço de estados tem poucos nós folha e um grande número de nós filho.

d)
(3,3,1)~>(2,2,0)~>(3,2,1)~>(3,0,0)~>(3,1,1)~>(1,1,0)~>(2,2,1)~>(0,2,0)~>(0,3,1)~>(0,1,0)~>(1,1,1)~>(0,0,0)

e) Para este caso específico creio que há necessidade, devido ao fato do algoritmo realizar uma busca em profundidade e o número de estados repetidos visitados não ser pequeno, ele poderia acabar entrando em loop.

Desafio 2:

h1(n) = número de quadrados em uma posição errada.

h2(n) = soma das distâncias que separam os quadrados das posições finais.

O número de blocos nada mais é que uma variável na resolução do algoritmo, ou seja, as heurísticas e a implementação não são afetados pelo número de blocos. Visto isso segue os estados gerados das heurísticas

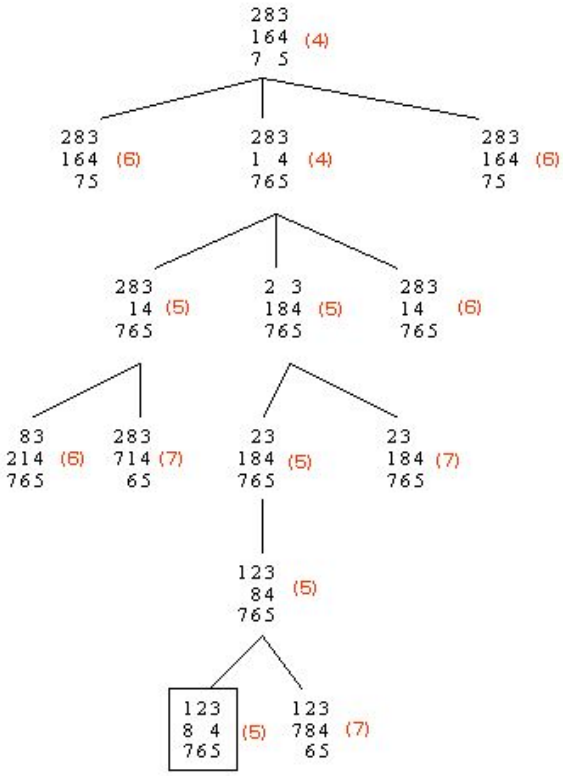
Explicação:

Seja h2(n) a função de maior valor, quer dizer, h2(n) domina h1(n), e f* o custo da solução ótima. O algoritmo A* vai visitar todos os nodos que tem valor f(n) =< f*. Toda vez que um nodo será expandido com h2(n), ele será também com h1(n) Pode existir um n por o qual o valor de h1(n) permitirá a expansão de um nodo e o valor de h2(n) faria a função f(n) ultrapassar o custo do caminho ótimo. Nesse caso, o nodo nunca seria expandido com a função h2(n). Então, menos nodos serão expandidos com a função h2.

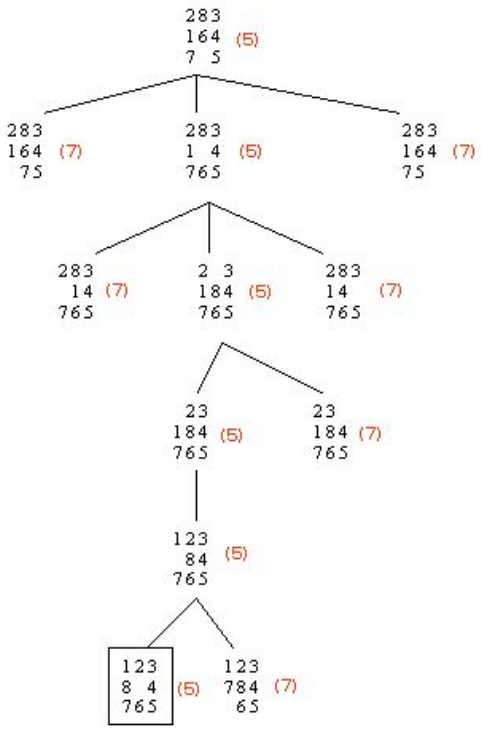
Na página seguinte podemos perceber que o que foi dito até agora na explicação de fato acontece com a h1;

Ilustração da explicação:

H1)



h2)


283
164 (5)
7 5
283
164 (7)
75
283
1 4 (5)
765
283
164 (7)
75
283
14 (7)
765
2 3
184 (5)
765
283
14 (7)
765
23
184 (5)
765
23
184 (7)
765
123
84 (5)
765
123
8 4 (5)
765
123
784 (7)
65