Inteligência Artificial

Segunda Lista de Exercícios

Arthur do Prado Labaki - 11821BCC017

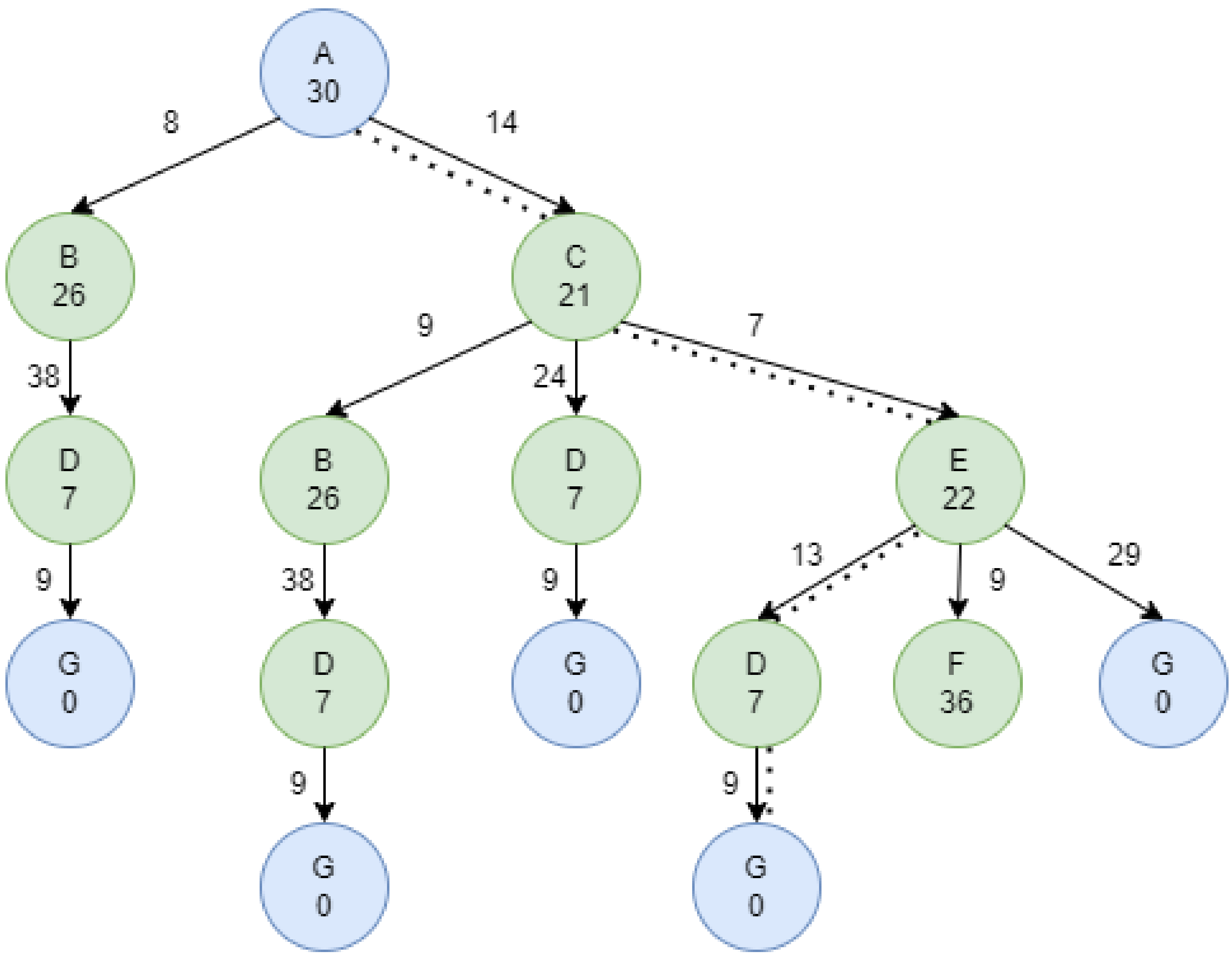
23-10, 2022

GBC063

Resolução do item 1)

Resolução do item A)

Dado o grafo dirigido, a sua respectiva árvore de busca é:

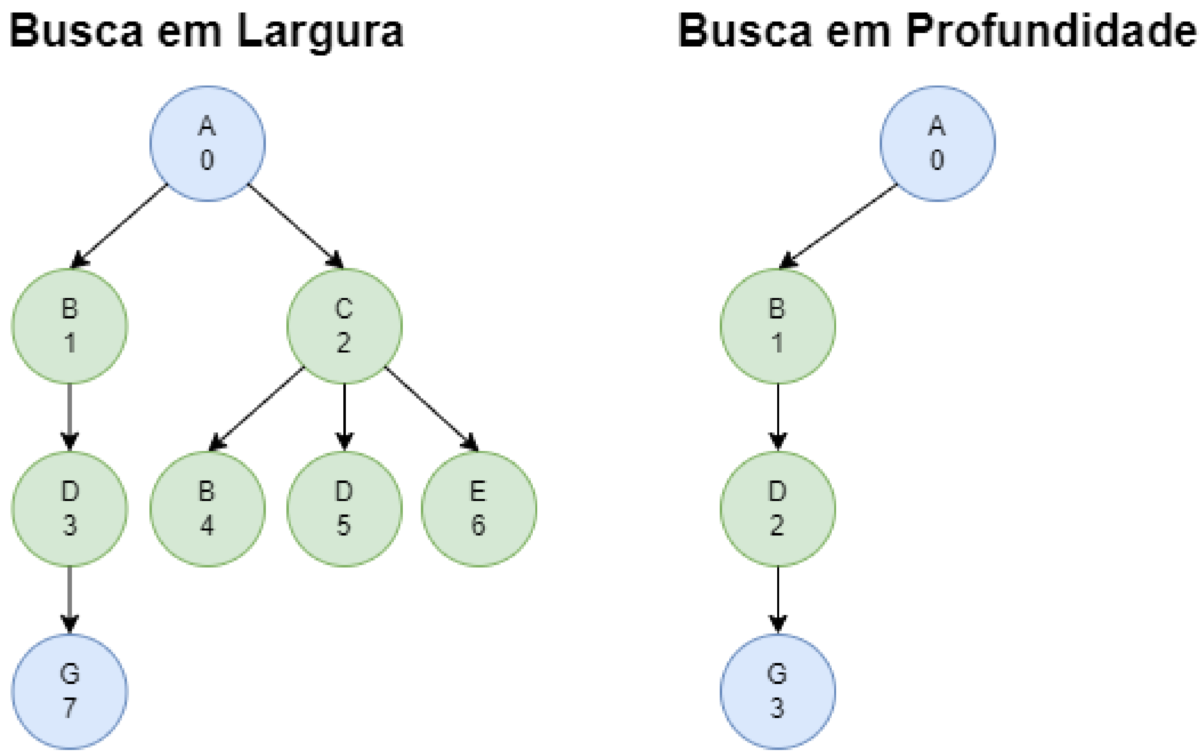


Resolução do item B)

O caminho ótimo (melhor caminho) de A para G é o A → C → E → D → G (caminho tracejado), com o menor custo possível de 43.

Resolução do item C)

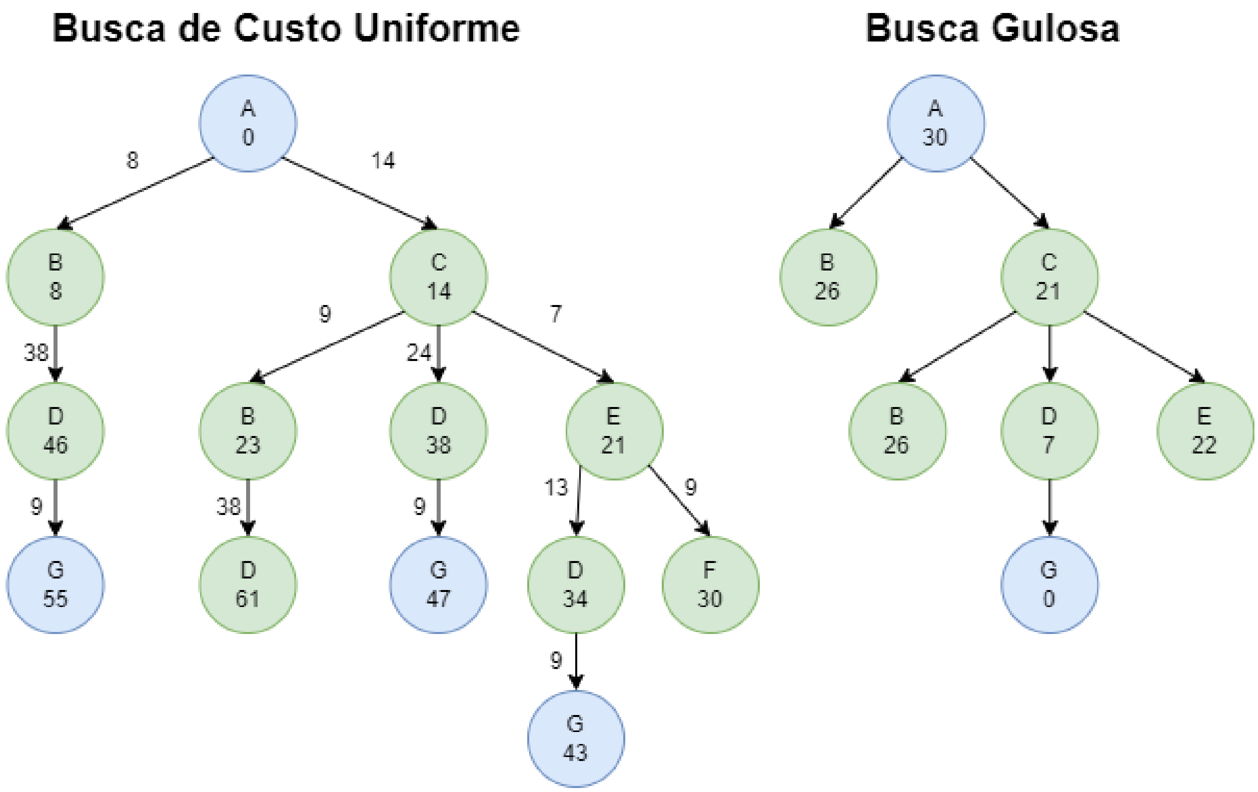
Utilizando as estrategias de busca, temos:



Como essas buscas são não informadas, o custo entre estados não é considerado. O número abaixo do estado representa o número do passo, começando em 0 (Ex: começa no passo 0, indo para o 1, 2, 3, ... até chegar em G).

Como a busca de custo uniforme leva em consideração o custo, ele foi mantido na árvore. Abaixo do nome do estado, tem a soma do custo anterior com o atual (Ex: A para $B = 8$ / B para $D = 8+38 = 46$ / D para $G = 45 + 9 = 55$ /...). A solução é a solução ótima, que é $A \rightarrow C \rightarrow E \rightarrow D \rightarrow G$.

A busca gulosa considera a função heurística, que já foi informada pelo texto, no qual é o numero abaixo do nome do estado. Vale relembrar que essa busca sempre avança para o próximo estado de menor função heurística, sem soma-las.

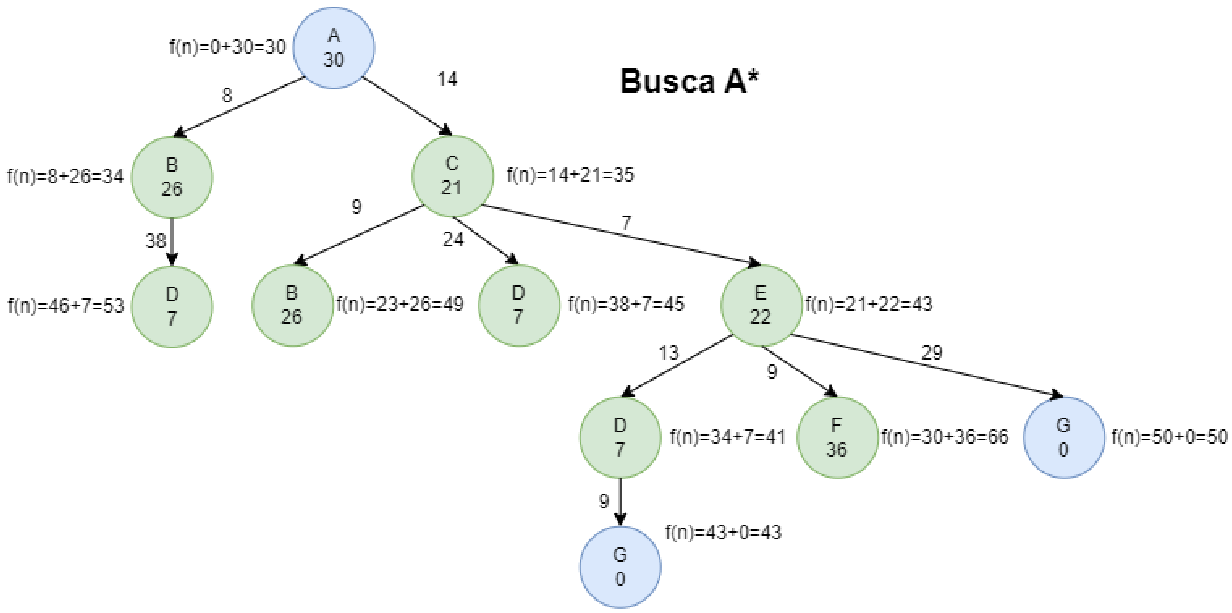


Por fim temos a busca A*, que utiliza a fórmula f(n) = g(n) + h(n), em que g(n) é o custo para ir ao estado n e h(n) é a função heurística para o estado n. Resumindo, é utilizado a soma das propriedades da busca de custo uniforme e da busca gulosa. Com isso, temos que a solução é A $\rightarrow$ C $\rightarrow$ D $\rightarrow$ G.

Resolução do item 2)

Supondo que é preciso colorir um mapa plano utilizando apenas quatro cores, de tal modo que não haja duas regiões adjacentes com a mesma cor, sabemos que:

- **Estado Inicial:** É o mapa utilizado com nenhuma região colorida;
- **Teste de objetivo:** Verifica se o mapa tem todas as regiões coloridas, utilizando apenas quatro cores, e que não há nenhuma região adjacente com a mesma cor;
- **Função sucessor:** Colorir alguma região do mapa que ainda não tenha cor;
- **Função de custo:** Número total de atribuições realizadas.



Resolução do item 3)

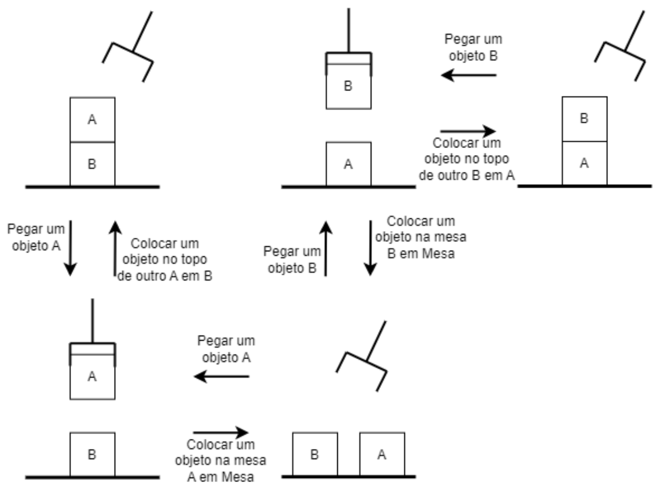
Considere o "mundo dos blocos" tendo um estado inicial e um estado meta como o explicitado. Assim, temos:

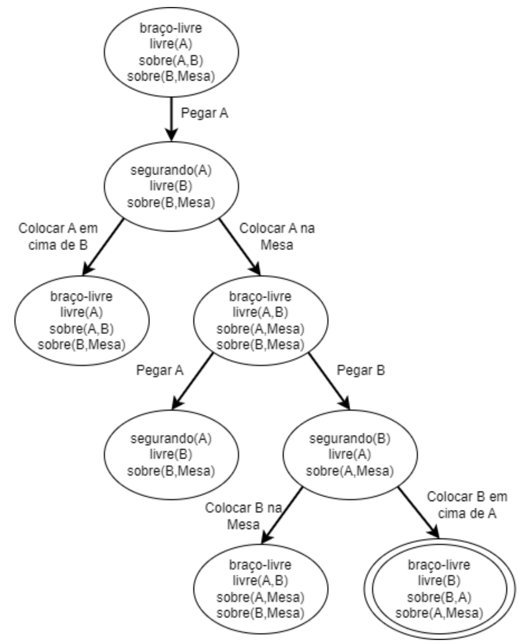
Estado Inicial e Estado de Meta

Estado Inicial: Braço-livre, livre(A), sobre(A,B) e sobre(B,Mesa)
Estado de Meta: Braço-livre, livre(B), sobre(B,A) e sobre(A,Mesa) (basicamente os blocos invertidos)

Espaço de Busca

Dado o exercício, é possível obter o espaço de busca em formato de grafo ou em formato de arvore, de maneiras diferentes. Eles são:





Caminho ótimo

O caminho ótimo, ou o melhor caminho para o problema é:

- Pegar bloco A;
- Colocar bloco A na Mesa;
- Pegar bloco B;
- Colocar bloco B em cima do bloco A;

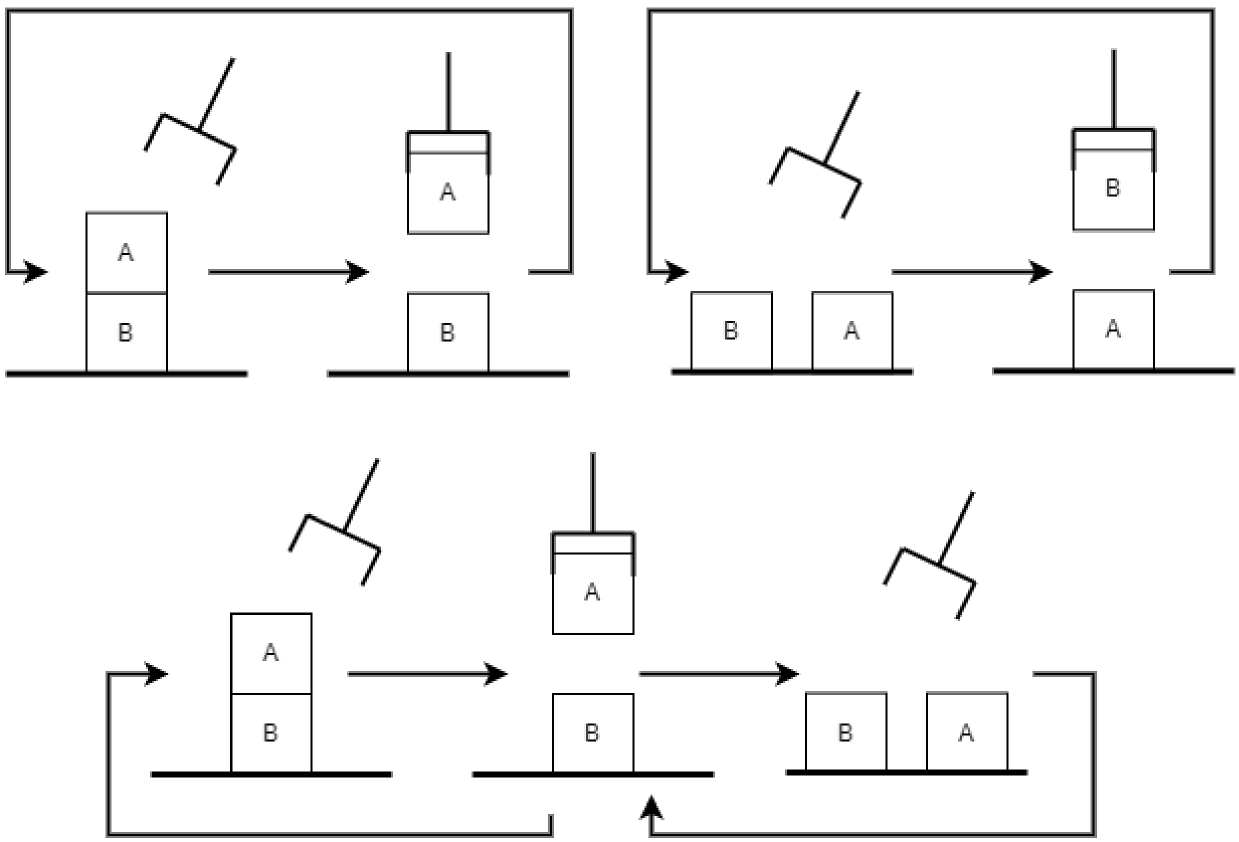
Ciclos de Repetição

É possível identificar diversos ciclos de repetição, que podem ser observados no desenho do grafo com imagens. Nele os ciclos se baseiam em caminhos entre estados que são de ambos os sentidos, como:

1. Pegar bloco A e colocar bloco A em cima de B e repetir esse ciclo;
2. Pegar bloco A, colocar bloco A na Mesa, pegar bloco B e colocar bloco b na Mesa e repetir essas duas ultimas ações;

3. Pegar bloco A, colocar bloco A na Mesa e repetir esse ciclo;

Para melhorar o entendimento, foi criado três ciclos.



Melhor Busca não Informada

Dentre busca em largura ou busca em profundidade, é possível observar que, analisando a árvore criada, a busca em profundidade com prioridade de direita conseguiria chegar na solução em apenas 4 transições. Já na busca em largura também com prioridade de direita, seria necessário 6 transições para alcançar o resultado. Logo a busca em profundidade seria melhor para esse tipo de problema, pois a árvore é mais vertical do que horizontal.

Resolução do item 4)

Dado o problema dos missionários e canibais, temos:

Resolução do item A)

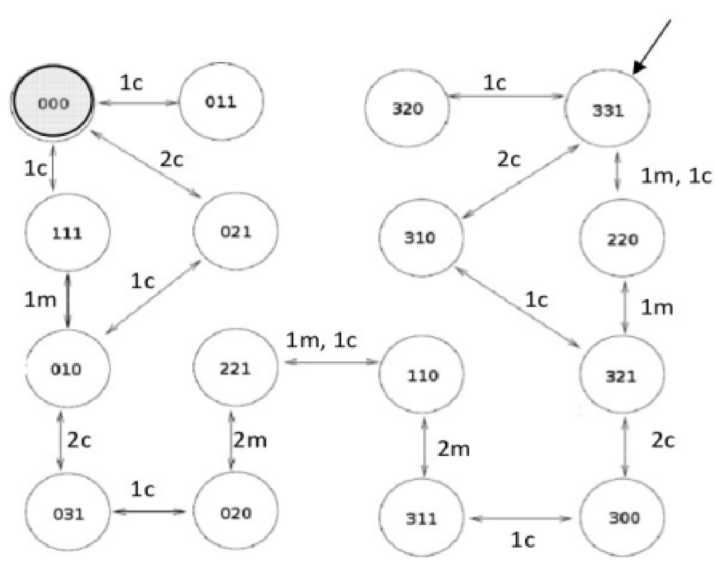
É possível representar esse problema na forma de uma árvore, no formato de (m,c,b), sendo m o número de missionários na margem inicial, c o número de canibais na margem inicial e b o lado do barco (1 para margem inicial e 0 para a outra margem).

Resolução do item B)

Sendo essa a estrutura dos dados, temos:

- **Estado inicial:** (3,3,1)
- **Estado de meta:** (0,0,0)
- **Operadores:** (3,2,0), (1,1,0), (1,0,1), (0,3,1), entre outros

Com isso, temos a estrutura completa:



Resolução do item 5)

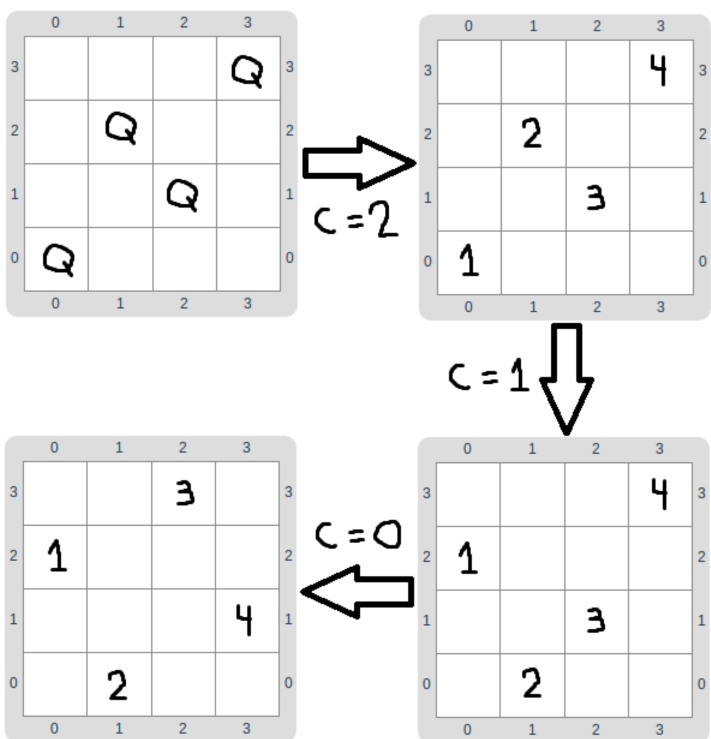
Para o problemas da 4 rainhas, temos:

Resolução do item A)

Sendo um tabuleiro 4x4 e cada rainha em uma coluna, com movimentação de rainhas trocando as posições das fileiras de duas rainhas, existem 6 vizinhos possíveis para um estado, pois cada rainha pode trocar de posição com outras 3, mas trocas A-B e B-A são as mesmas (Supondo rainhas A, B, C, D, temos que a rainha A pode trocar com B, C, D, a rainha B com C, D, a rainha C com D, assim nenhuma se repete).

Resolução do item B)

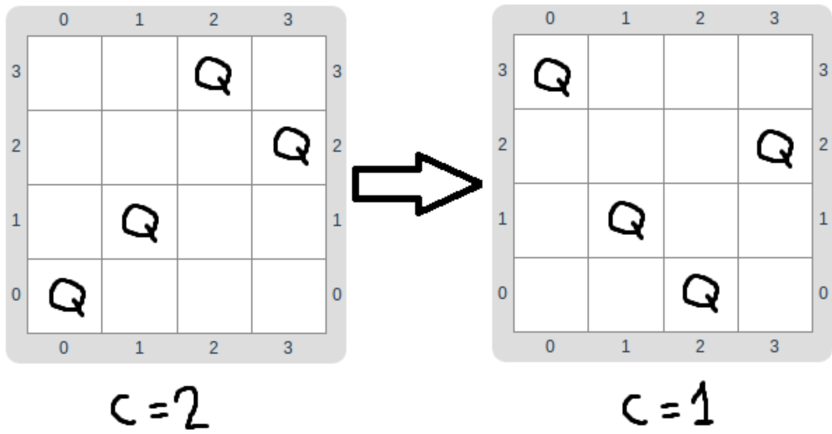
Usando o algoritmo de subida de colina, temos que as etapas de execução são:



Vale lembrar que o C é o número de colisão entre as rainhas.

Resolução do item C)

Dado tabuleiro e utilizando subida de encosta, temos que o custo do estado atual é de 2. Não é um ótimo local, pois existe possibilidade de troca de rainhas, como a x0 pela x3, resultando no tabuleiro:



Resolução do item D)

De acordo com o problema formulado, é impossível que essa configuração seja realizada, pois é presumido que exista uma rainha por coluna no tabuleiro. Porem,caso essa configuração exista, seu custo seria de 4 e seria um ótimo local, pois qualquer movimento resultaria no mesmo tabuleiro, porem não é um ótimo global, pois ainda existe colisões entre rainhas.

Resolução do item E)

Sendo um tabuleiro 4x4 e cada rainha em uma coluna, com movimentação de qualquer rainha para outra casa na mesma coluna, existem 12 , pois para cada rainha existe 3 possíveis movimentos.

Resolução do item F)

Dado esse tabuleiro e essa nova movimentação, seu custo atual é de 2 e esse é um ótimo local, pois qualquer outro possível movimento resulta em um tabuleiro com custo igual ou maior ao atual, porem não é um ótimo global, pois ainda existe colisão entre rainhas (custo não é 0).

Resolução do item 6)

Preenchendo a tabela, temos:

custo atual	próximo. custo	ΔE	p(T=100)	p(T=50)	p(T=10)
50	40	-10	1.105	1.221	2.718
50	100	50	0.606	0.367	0.006
50	200	150	0.223	0.049	3.059E-7

ΔE = Valor(Prox nó) - Valor(Nó atual)

p = $e^{-\Delta E/T}$

Com isso podemos observar que:

- A probabilidade de se mudar para um vizinho com um custo menor é sempre maior ou igual a 100% (sempre maior que 1);
- Sendo um vizinho de maior custo, a medida que a temperatura diminui, a probabilidade de se mudar para esse pior vizinho diminui(diretamente proporcional);
- À medida que a diferença entre o custo do vizinho e o custo do estado atual aumenta, a probabilidade de se mudar para esses vizinhos diminui (inversamente proporcional, maior delta E, menor probabilidade).