7.16.

a) $a \notin A'$ e $w'(a) > w(a)$.

Não há nada a fazer. Essa aresta não será incorporada à AGM.

b) $a \notin A'$ e $w'(a) < w(a)$.

A aresta é candidata a entrar na AGM. Considerando que a aresta a conecta os vértices i e j, encontramos a maior aresta b no caminho entre i e j definido pela AGM original. Se $w(b) > w'(a)$, então a aresta b pode ser removida e a aresta a entra em seu lugar na AGM.

c) $a \in A'$ e $w'(a) > w(a)$.

A aresta a conectando os vértices i e j é removida da árvore e verificamos qual é a aresta de menor peso que liga as duas árvores resultantes T_i e T_j, que é adicionada, restaurando a AGM. Há $O(v^2)$ arestas possíveis a serem avaliadas.

d) $a \in A'$ e $w'(a) < w(a)$.

Não há nada a fazer. Essa aresta vai continuar na AGM.

7.17.

a) Inicialmente, calcule os caminhos mínimos entre todos os pares de cidades e armazene os valores em uma tabela. O desafio é como incorporar os custos de estadia, uma vez que o nosso objetivo não é minimizar a distância percorrida, mas os custos de estadia. Entretanto, nem todas as rotas são viáveis e o modelo deve refletir isso.

O modelo é um grafo acíclico direcionado H com $n \times (m+1)$ vértices. Os vértices são rotulados como v_{ip} para $i \in \{1, 2, \ldots, u\}$ e $p \in \{0, 1, \ldots, m\}$. Cada nó v_{ip} corresponde a opção de pernoitar na cidade i no dia p.

Para qualquer $i, j \in \{1, 2, \ldots, n\}$ em que $i \neq j$ e $p \in \{1 \ldots m\}$, adicione a aresta $(v_{i(p-1)}, v_{jp})$ no grafo H se $d(i,j) < u(p)$. Isso significa que a mercadoria pode ser transportada da cidade i para a cidade j sem exceder o limite diário. Mais ainda, para cada vértice v_{ip}, atribua o custo c_i ao vértice.

b) O frete de menor custo é o caminho de v_{s0} a v_{tm} em que o custo total dos vértices é mínimo. Se v_{tm} não é alcançável a partir de v_{s0}, nenhum caminho que satisfaz os requisitos existe.

Podemos transformar o problema do caminho de custo mínimo dos vértices em um problema de caminho mínimo. Uma vez que o grafo seja direcionado, para cada aresta (u, v) podemos atribuir o custo do nó v como o peso da sua aresta. Isso reduz o problema em pauta para o problema do caminho mínimo, que pode ser resolvido pelo **algoritmo de Dijkstra**.

c) As distâncias mínimas entre as cidades podem ser calculadas por n execuções do algoritmo de Dijkstra, o que resulta em $O(n^3)$. O grafo H contem $O(nm)$ vértices e $O(n^2m)$ arestas, com custo $O(n^2m)$ para construir o grafo. Uma vez que H é um grafo acíclico não direcionado, o caminho mínimo em H pode ser calculado com custo $O(n^2m)$. Assim, o custo do algoritmo é $O(n^3 + n^2m)$.

d) Um caminho no grafo a partir de v_{s0} até v_{tm} corresponde ao trajeto com m dias de duração. Especificamente, cada aresta $(v_{i(p-1)}, v_{jp})$ ao longo do caminho especifica o transporte entre as cidades i e j no dia p. Note que a aresta somente estará no grafo H se não violar as restrições do problema. Mais ainda, como não há arestas entre $(v_{i(p-1)}, v_{ip}) \forall i, p \in H$, não há estadia na mesma cidade por duas noites consecutivas. A transformação do problema do caminho de menores custos nos nós no problema do caminho mínimo preserva o custo dos caminhos correspondentes. Portanto, o algoritmo de caminho mínimo no grafo transformado calcula o roteiro de melhor frete desejado.

7.18.

a) O problema é modelado como um grafo não direcionado ponderado em que os vértices são os pontos de armazenamento e arestas representam as conexões entre dois pontos de armazenamento. O peso atribuído a cada aresta representa a distância entre dois pontos.

b) Este problema pode ser resolvido por uma árvore geradora mínima. Os pontos de armazenamento deverão ser todos incluídos e não se espera que haja ciclo, logo é uma árvore geradora. Além disso desejamos minimizar a distância percorrida pela água, ou seja, a soma total das arestas. Logo, basta aplicar o algoritmo de Prim ou o algoritmo de Kruskal.

c) $O(A \log V)$ onde A representa o número de arestas e V o de vértices.

7.19.

a) O problema é modelado como um grafo direcionado poderado, em que cada vértice representa uma cidade e uma aresta de u para v representa o transporte do dinheiro entre u e v por algum agente. O peso de uma aresta (u,v) representa o valor cobrado para transferir o dinheiro de u para v. Como múltiplos agentes podem oferecer o transporte entre o mesmo par de cidades, múltiplos pesos podem ser atribuídos a mesma aresta. Visando minimizar a perda final, devemos escolher como peso de uma aresta o menor valor cobrado, rotulando a mesma com um identificador do respectivo agente para construção da solução final. O problema se reduz a determinar o caminho mais curto (menor perda total) entre as cidades onde Marcelo e Alice estão.

b) O problema pode ser resolvido pelo **algoritmo de Dijkstra** para determinar o menor caminho entre os vértices no grafo que correspondem às cidades envolvidas. As arestas selecionadas indicam qual agente será utilizado. O algoritmo é ótimo.

Capítulo 8

8.3.

a) O algoritmo BM original propõe duas heurísticas para calcular o deslocamento: (i) Heurística ocorrência (do inglês *ocurrence*): alinha a posição no texto