Teoria dos Grafos – COS 242
2022/2

Terceira Lista de Exercícios

ATENÇÃO! Para um melhor rendimento do processo de aprendizagem, responda às perguntas de forma precisa e concisa contemplando adequadamente cada questão.

Questão 1: Dê um exemplo de DAG com n vértices que admite apenas uma ordenação topológica. Dê um exemplo de um DAG com n vértices que admite pelo menos n ordenações topológicas distintas.

Questão 2: Modifique o algoritmo de Dijkstra apresentado em aula de modo a encontrar não somente a distância do vértice inicial a todos os outros, mas também o caminho mínimo. Escreva o pseudo-código do algoritmo modificado.

Questão 3: Considere o grafo ilustrado na figura abaixo. Utilizando uma tabela (conforme apresentado em aula), mostre o funcionamento do algoritmo de Dijkstra passo-a-passo. Utilize o vértice G como vértice inicial.

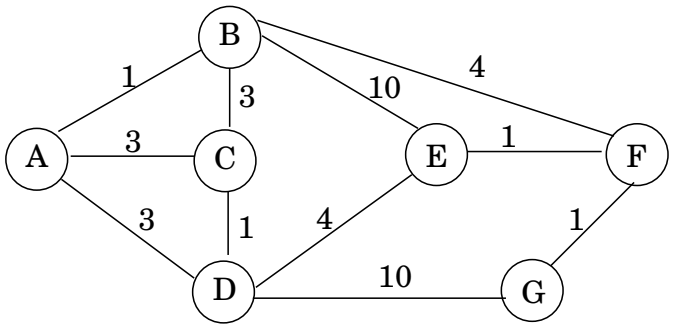

Figura 1: Grafo não-direcionado com pesos.

Questão 4: O algoritmo de Dijkstra assume que os pesos associados às arestas são sempre positivos. Esta premissa foi necessária para provar a corretude do algoritmo, como vimos em aula. Dê um exemplo de um grafo direcionado com pesos negativos para o qual o algoritmo de Dijkstra produz resultados errados. Por que a prova da corretude do algoritmo falha quando temos pesos negativos?

Questão 5: Qual é a principal diferença entre o algoritmo A* e o algoritmo de Dijkstra? Em particular, qual informação o algoritmo A* necessita e como isto é explorado na execução do algoritmo? Dê um exemplo concreto onde o algoritmo A* utilizando a devida informação realiza bem menos passos que o algoritmo de Dijkstra.

Questão 6: Considere um jogo de quebra-cabeça em um tabuleiro de 3x3 com peças enumeradas de 1 à 8 cujo objetivo é movimentar as peças para atingir uma configuração ordenada. A figura abaixo ilustra um possível estado inicial e o estado final desejado.

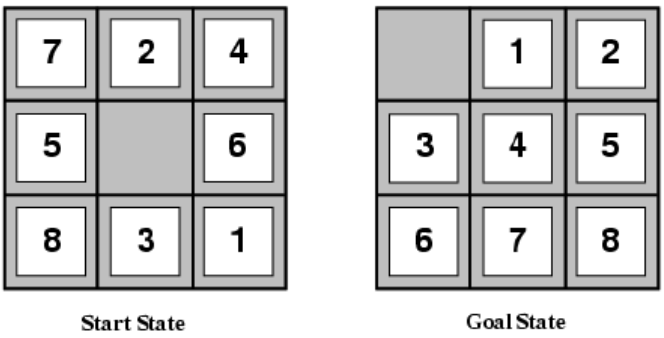

Figura 2: Quebra-cabeça de oito peças em tabuleiro 3x3.

Dado um estado inicial, você gostaria de encontrar o menor número movimentações necessárias para chegar ao estado final. Entretanto, encontrar a sequência mínim pode não ser computacionalmente viável. Dessa forma, descreva um algoritmo eficiente para encontrar sequências de movimentações que sejam curtas (não necessariamente mínimas), deixando claro as principais ideias e estruturas de dados.