Teoria dos Grafos - COS 242
2024/2

Quinta Lista de Exercícios

ATENÇÃO! Para um melhor rendimento do processo de aprendizagem, responda às perguntas de forma precisa!

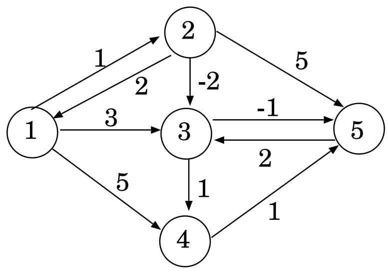
Questão 1: Considere o problema de determinar se um grafo é 2-colorível. Descreva (em pseudo-código) um algoritmo eficiente (de tempo linear) para este problema. Seu algoritmo recebe um grafo como entrada e retorna verdadeiro ou falso, de acordo. Analise a complexidade computacional do seu algoritmo. Dica: determine primeiro a relação entre um grafo bipartido e um grafo 2-colorível.

Questão 2: Considere o problema de coloração de vértices de um grafo e o algoritmo guloso visto em aula. Descreva (em pseudo-código) um algoritmo eficiente (linear no número de cores sendo usadas) para determinar a menor cor que pode ser atribuídada a um vértice, onde parte dos vizinhos estão coloridos. Seu algoritmo recebe como entrada um grafo, um vértice v, e uma coloração parcial dos vértices, e retorna a menor cor para v. Analise a complexidade computacional do seu algoritmo indicando as estruturas de dados sendo utilizadas.

Questão 3: Considere o problema da soma do subconjunto e o algoritmo apresentado em aula que constroi a matriz $M[i, w]$. Descreva (em pseudo-código) um algoritmo eficiente que recebe como entrada a matriz M preenchida e gera o conjunto ótimo de objetos. Determine a complexidade do seu algoritmo.

Questão 4: Considere a função recursiva utilizada pelo algoritmo de Floyd-Warshall, $d(i, j, k)$. Responda as perguntas abaixo:

1. Defina exatamente o significado do valor da função $d(i, j, k)$, assim como seus parâmetros.
2. Considerando o grafo abaixo, calcule $d(2, 5, k)$ e $d(1, 4, k)$ para $k = 0, 1, \ldots, 5$.

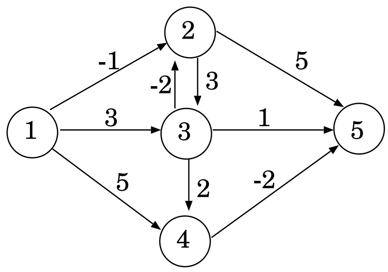


Questão 5: Considere o algoritmo de Floyd-Warshall apresentado em aula. Modifique o pseudo-código do algoritmo para que o mesmo obtenha também o caminho mínimo (sequência de vértices) entre dois vértices i e j quaisquer. Repare que você deve criar e manter uma estrutura de dados auxiliar para obter esta informação, similar à utilizada pelo algoritmo de Dijkstra.

Questão 6: Considere a função recursiva utilizada pelo algoritmo de Bellman-Ford, $OPT(i, v)$. Responda as perguntas abaixo:

1. Defina exatamente o significado do valor da função $OPT(i, v)$, assim como seus parâmetros.

2. Considerando o grafo abaixo, assuma que $t = 5$ (vértice destino), e calcule $OPT(i, 1)$ e $OPT(i, 3)$ para $i = 0, 1, \ldots, 4$.



Questão 7: Vimos duas melhorias práticas para o algoritmo de Bellman-Ford apresentadas em aula: Uma para reduzir a quantidade de memória e outra para a reduzir o tempo de execução do algoritmo. Descreva o algoritmo (em pseudo-código) que implementa estas duas melhorias.

Questão 8: Considere o problema de detectar um ciclo negativo em um grafo direcionado com pesos. Lembrando que um ciclo negativo é um ciclo no grafo tal que a soma dos pesos das arestas do ciclo é menor do que zero. Mostre como podemos utilizar o algoritmo de Bellman-Ford para detectar a presença de um ciclo negativo.

Questão 9: Calcule o tamanho do maior emparelhamento para cada um dos grafos abaixo:

1. Estrela com k arestas.
2. Caminho com k arestas.
3. Ciclo com k arestas.
4. Grafo completo com n vértices, K_n.
5. Grafo bipartido completo com partes de tamanho n_1 e n_2.
6. Grafo de Petersen (um famoso grafo, ilustrado na figura abaixo).