MCTA003-17 - Análise de Algoritmos
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC
Prof. Maycon Sambinelli

Grafos

1. Dado um grafo G, o *quadrado* de G é o grafo G^2 tal que $V(G^2) = V(G)$ e, para cada par de vértices $x, y \in V(G)$, a aresta xy vai existir em $E(G^2)$ se e somente se temos $xy \in E(G)$ ou se existe algum outro vértice $w \in V(G)$ tal que as arestas xw e wy existem em $E(G)$. Em outras palavras, em G^2 existe aresta entre todo par de vértices cuja distância é 1 ou 2 em G.

Descreva um algoritmo que recebe um grafo G qualquer e devolve G^2. Tenha certeza que o grafo G^2 não possui arcos paralelos. Analise o tempo de execução se o algoritmo for implementado usando listas de adjacências e se for usando matriz de adjacências. Não é preciso apresentar uma demonstração de correção.

2. O digrafo transposto de um digrafo D é o digrafo D^T tal que $V(D^T) = V(T)$ e $E(D^T) = \{vu \colon uv \in E(D)\}$. Assim, D^T é D com todas as arestas invertidas. Descreva um algoritmo que recebe um digrafo D qualquer e devolve D^T. Analise o tempo de execução se o algoritmo for implementado usando listas de adjacências e se for usando matriz de adjacências. Não é preciso apresentar uma demonstração de correção.

3. Adapte o algoritmo da Busca em Largura para que o mesmo conte o número de componentes conexas de um grafo e analise a complexidade do algoritmo resultante.

4. Leia a Seção 24.2 das notas de Aula dos Professores Carla N. Lintzmayer e Guilherme O. Mota (veja seção de referências bibliográficas do curso) para aprender sobre o Algoritmo de Busca em Profundidade.

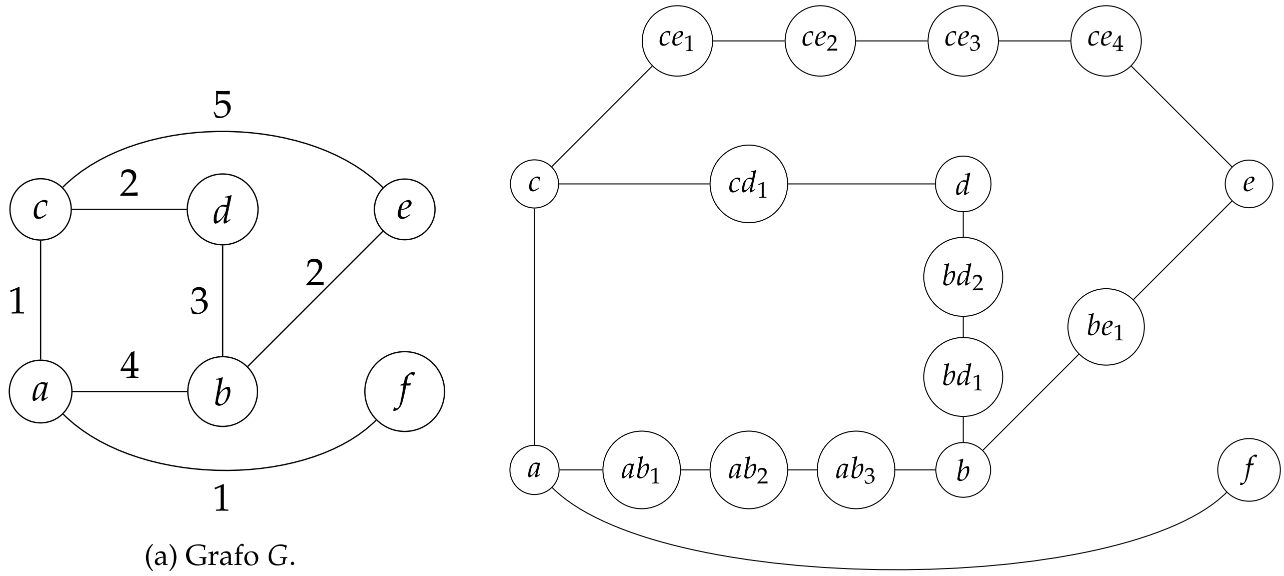
5. Considere o grafo G definido por $V(G) = \{a, b, c, d, e, f, g, h, i\}$ e $E(G) = \{ad, de, ea, ba, bf, fg, gb, cb, ch, hi, ic\}$. Execute a busca em largura sobre G a partir do vértice e. Execute a busca em profundidade sobre G a partir do vértice e. Você pode representar a execução mostrando a evolução do preenchimento de dois vetores, `visitado` e `predecessor`, que são indexados pelos vértices.

6. Prove eu apresente um contraexemplo para a seguinte afirmação: da mesma forma que acontece com a árvore de busca resultante da execução do algoritmo BFS, podemos usar a árvore de busca resultante do algoritmo DFS para encontrar caminhos mínimos da raiz até todos os outros vértices do grafo.

7. Escreva um algoritmo que recebe um grafo G e dois vértices s e v e devolve a sequência de vértices de um sv-caminho em tempo $O(|V(G)| + |E(G)|)$.

8. Dada uma árvore T, o *diâmetro* de T é o valor da maior distância entre quaisquer dois vértices de T. Formalmente, é o número $\max\{dist(u, v) \colon u, v \in V(T)\}$, onde $dist(u, v)$ é a distância entre u e v em T, em número de arestas. Escreva um algoritmo que, dada T, determine o diâmetro de T em $O(V)$. Justifique a correção do seu algoritmo.

9. Seja $G = (V, E)$ um grafo e $w \colon E(G) \to \mathbb{Z}_+$ uma função de custo nas arestas tal que $w(e) \geq 1$. Construa o grafo H tal que cada aresta $e \in E(G)$ é substituída por um caminho com $w(e)$ arestas sem custo em H (ou, similarmente, de custo 1). Assim, H possui os mesmos vértices de G e alguns vértices extras. Veja as Figuras 1a e 1b.



(a) Grafo G.

(b) Grafo H.

Figura 1: Exemplos para o exercício 9.

Dado um vértice $s \in V(G)$, qual é o tempo de execução da busca em largura sobre H a partir de s? É possível escrever esse tempo em função de $|V(G)|$ e $|E(G)|$? Para todo $v \in V(G)$, o custo do sv-caminho mínimo em H encontrado pela busca em largura é o mesmo do sv-caminho mínimo em G? Formalize devidamente sua resposta.