

MAC 5711 - Análise de Algoritmos

Departamento de Ciência da Computação

Segundo semestre de 2019

Lista 7

1. **(CRLS Ex. 22.2-1)** Simule o funcionamento da BFS no grafo da Figura 22.2(a) do CLRS (terceira edição) a partir do vértice 3, determinando os valores de d e π para cada vértice.
2. **(CRLS Ex. 22.2-2)** Simule o funcionamento da BFS no grafo da Figura 22.3 do CLRS (terceira edição) a partir do vértice u , determinando os valores de d e π para cada vértice.
3. **(CRLS Ex. 22.2-5)** Argumente que o valor de $d[u]$ atribuído ao vértice u na BFS é independente da ordem em que os vértices das listas de adjacências são dados. Por outro lado, mostre, usando o exemplo da Figura 22.3 do CLRS, que a árvore BFS depende da ordem dos vértices nas listas de adjacências.
4. **(CRLS Ex. 22.2-6)** Dê um exemplo de um grafo dirigido $D = (N, A)$, um vértice s em D , e um conjunto $A_\pi \subseteq A$ de arcos em D que formam uma árvore tal que, para todo vértice u em D , o único caminho entre s e u em A_π é um caminho mínimo em D entre s e u , porém A_π jamais seria produzida por uma execução da BFS em D , independente da ordem dos nós nas listas de adjacências de D .
5. Escreva uma versão não recursiva da busca em profundidade.
6. Execute uma busca em profundidade a partir do vértice 0 no grafo dirigido dado pelas listas de adjacência a seguir. Exiba o rastreamento da busca.

```
0: 1 4
1: 2 5
2: 3
3: 7
4: 8
5: 4
6: 5 10 2
7: 11 6
8: 9
9: 5 8
10: 9
11: 10
```

7. **(CRLS Ex. 22.3-1)** Desenhe uma tabela 3×3 , com as linhas e colunas indexadas pelas cores branco, cinza e preto. Em cada entrada (i, j) , indique se, em qualquer ponto durante uma DFS de um grafo dirigido, pode existir um arco de um nó de cor i para um nó de cor j . Para cada arco possível, indique as classificações que ele pode ter (de árvore, de retorno, para frente, cruzado). Faça um segundo quadro considerando um grafo não dirigido.
8. **(CRLS Ex. 22.3-2)** Mostre como a DFS funciona no grafo da Figura 22.6 do CLRS (segunda edição). Assuma que o laço das linhas 5-7 da DFS visitam os vértices em ordem alfabética, e que os vértices se encontram em ordem alfabética nas listas de adjacências. Mostre os valores de d e f para cada vértice ao final da DFS.
9. **(CRLS Ex. 22.3-7)** Mostre um contra-exemplo para a conjectura que se existe um caminho de u a v em um grafo dirigido G , e se $d[u] < d[v]$ numa DFS de G , então v é descendente de u na floresta DFS produzida.

10. **(CRLS Ex. 22.3-8)** Mostre um contra-exemplo para a conjectura que se existe um caminho de u a v em um grafo dirigido G , então qualquer DFS deve resultar em $d[v] \leq f[u]$.
11. **(CRLS Ex. 22.3-10)** Mostre como um vértice u num grafo dirigido pode terminar sozinho numa árvore de uma floresta DFS mesmo tendo arcos saindo e entrando dele em G .
12. **(CRLS Ex. 22.3-12)** Um grafo dirigido G é *unicamente conexo* se existe no máximo um caminho (dirigido) de u para v para todo par de vértices u e v de G . Dê um algoritmo eficiente para determinar se G é unicamente conexo.
13. Escreva uma generalização comum das buscas em largura e em profundidade. Sua função deve usar uma estrutura de dados auxiliar que pode operar como fila ou como pilha. Se a estrutura operar como fila, a função executa busca em largura, e se operar como pilha, a função executa busca em profundidade.
14. Escreva um algoritmo que decida se um grafo é conexo. Analise o seu consumo de tempo.
15. Escreva um algoritmo que determine o número de componentes conexas de um grafo. Analise o seu consumo de tempo.
16. Um grafo $G = (V, E)$ é *bipartido* se seu conjunto de vértices V pode ser bipartido em dois conjuntos disjuntos de vértices A e B e toda aresta de E tem uma ponta em A e outra em B . Escreva um algoritmo que, dado um grafo, determine se o grafo é ou não bipartido. Analise o seu consumo de tempo.
17. Dada uma árvore $T = (V, E)$, o *diâmetro* de T é o número $\max\{d(u, v) : u, v \in V\}$, onde $d(u, v)$ é a distância entre u e v em T . Escreva um algoritmo que, dado T , determine o diâmetro de T . Analise o seu consumo de tempo.