# MAC 5711 - Análise de Algoritmos

*Departamento de Ciência da Computação*

Segundo semestre de 2018

## Lista 6

1. Escreva uma versão não recursiva da busca em profundidade.

2. 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
```

3. **(CRLS Ex. 22.3-1)** Desenhe uma tabela $3 \times 3$, 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.

4. **(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.

5. Considere as seguintes conjecturas:

   (a) **(CRLS Ex. 22.3-7)** Se existe um caminho de $u$ a $v$ em um grafo dirigido $G$, e se $u.d < v.d$ numa DFS de $G$, então $v$ é descendente de $u$ na floresta DFS produzida.

   (b) **(CRLS Ex. 22.3-8)** Se existe um caminho de $u$ a $v$ em um grafo dirigido $G$, então qualquer DFS deve resultar em $v.d \leq u.f$.

   (c) **(CRLS Ex. 22.3-10)** Se um vértice tem arcos entrando e saindo dele, então em qualquer DFS no grafo a componente da árvore DF que o contém tem mais de um vértice.

   Mostre que as três são falsas, apresentando contraexemplos.

6. **(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.

7. 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.

8. Escreva um algoritmo baseado em DFS que determine o número de componentes conexas de um grafo. Analise o seu consumo de tempo.

9. 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 $T$. A seu critério, você pode supor que $T$ é dado como um grafo, ou como uma estrutura de dados, com raíz, filhos etc. Explique sucintamente sua suposíção. Analise o seu consumo de tempo.

10. O *grau de entrada* $u.g$ de um vértice $u$ de um digrafo é o número de arcos que terminam em $u$.

    (a) Mostre que se um digrafo é acíclico, ele tem um vértice com grau de entrada 0.

    (b) Considere o seguinte algoritmo:
    **enquanto** *$G$ não é vazio*
            *Encontre um vértice de grau de entrada 0 e remova de $G$*
    **devolva** *os vértices numerados na ordem em que foram removidos*

    Mostre que se o grafo dado é acíclico, o algoritmo devolve uma ordenação topológica.

    (c) Suponha $G$ dado por listas de adjacências. Implemente o algoritmo acima em pseudocódigo, da forma mais eficiente que puder. Analise a complexidade.