Lista 1 - Algoritmos para Grafos

Prof. Alex Martins – UVA

1. Consistência.  Escreva uma função que confira a consistência da representação de um digrafo G em relação a quantidade de arcos.  (Em particular, verifique se a representação não tem [laços](http://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/digraphs.html" \l "loop) nem [arcos paralelos](http://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/digraphs.html" \l "parallel).)
2. Implemente a função DIGRAPHpath passando o digrafo abaixo e verifique se há caminho de 0 para 5. E de 5 para 2?
3. Escreva uma função DIGRAPHdestroy que destrua a representação de um digrafo G, ou seja, libere o espaço que a representação ocupa na memória. Escreva uma versão para digrafos representados por vetor de arcos, outra para digrafos representados por matriz de adjacência, e outra para digrafos representados por listas de adjacência.
4. Digamos que um arquivo é *gráfico* se sua primeira linha contém um inteiro V e cada uma das demais linhas contém dois inteiros pertencentes ao intervalo 0..V-1.  Eis um exemplo:

7

0 1

0 5

1 0

1 5

2 4

3 1

(É claro que o último caractere do arquivo é um \n.)  Se interpretarmos cada linha do arquivo como um arco, podemos dizer que o arquivo define um digrafo com vértices 0..V-1.  Escreva uma função DIGRAPHconstruct que receba um arquivo gráfico e construa uma representação do digrafo. Use as funções DIGRAPHinit e DIGRAPHinsertA.

1. Escreva uma função que gere um dígrafo aleatório. A função DIGRAPHaleatorio(G, V, A), devolverá um Digrafo G com V vértices e A arcos criados aleatoriamente (usar função rand() ).