

Lista 1 - Algoritmos para Grafos – 16/11/2015
Prof. Alex Martins – UVA
Entregar: 23/11/2015

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](#) nem [arcos paralelos](#).)
2. Escreva uma função que construa o [grafo](#) de [Petersen](#). (Pesquise sobre grafo de Petersen na internet).
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`.
5. 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()`).