



ALGORITMOS E ESTRUTURAS DE DADOS  
AULA DE LABORATÓRIO #06 - GRAFOS

## Objectivos

Neste laboratório abordam-se os tópicos da representação e manipulação de grafos representados por listas de adjacências.

## Plano da Aula

A tabela abaixo mostra um exemplo de duas representações de um grafo. Este grafo é ponderado e não direccionado, com  $V = 6$  vértices e  $E = 5$  arestas; tem densidade  $2E/V = 1.666...$  e grau máximo 3. Os vários formatos são especificados em mais detalhe no texto das questões.

Vector de arestas	Lista de adjacências
6 5	6
0 1 2	1:2 2:1 3:3 -1
0 2 1	0:2 -1
0 3 3	0:1 3:4 4:5 -1
2 3 4	0:3 2:4 -1
2 4 5	2:5 -1
	-1
grafo6.edge	grafo6.ladj

Neste problema pretende-se fazer uso da representação de grafos em lista de adjacências. Complete o código do programa `ev2e1` de forma a ler de um ficheiro do tipo `.edge`, como especificado acima, as arestas de um grafo não direccionado, ponderado, a implementar as seguintes tarefas:

1. Constrói, a partir dos dados lidos, uma estrutura de dados que representa o grafo, usando a representação em lista de adjacências.
2. Escreve um ficheiro de saída, com a extensão `ladj`,

`<nome_grafo>.ladj`

com uma representação em texto da lista de adjacências. A primeira linha especifica o número de vértices do grafo; cada linha seguinte diz respeito a um vértice, por ordem, e indica a sua lista de vértices adjacentes. Para cada adjacência são dados o vértice e o peso da aresta, unidos por `' : '`. A linha termina com `-1`. Quando um vértice é isolado, a sua linha existe para manter a ordenação, mas contém apenas `-1` (ver exemplo na figura acima).

3. Calcula, percorrendo a representação em lista de adjacências do grafo, o grau de cada nó e a densidade média do grafo, escrevendo esses valores para `stdout`.
4. Dado um nó de partida, lido de `stdin`, faz um varrimento em largura (BFS - *Breadth First Search*) do grafo imprimindo todos os nós de acordo com a sua distância (medida em número de arestas) ao nó inicial.

Nos ficheiros `LinkedList.h` e `LinkedList.c` encontram-se definidos os protótipos para manipulação de uma lista simplesmente ligada de `Items` e as suas implementações, respectivamente, que deverá usar no seu programa. O tipo `Item` é definido no ficheiro `defs.h`. Da mesma forma nos ficheiros `queue.h` e `queue.c` encontram-se definidos os protótipos para implementação de uma fila simples (sem prioridades!) de `Items` e as suas implementações, respectivamente, que deverá usar no seu programa.

O programa, depois de completado, deve ser invocado com o comando

```
ev2el <nome_grafo>.edge
```