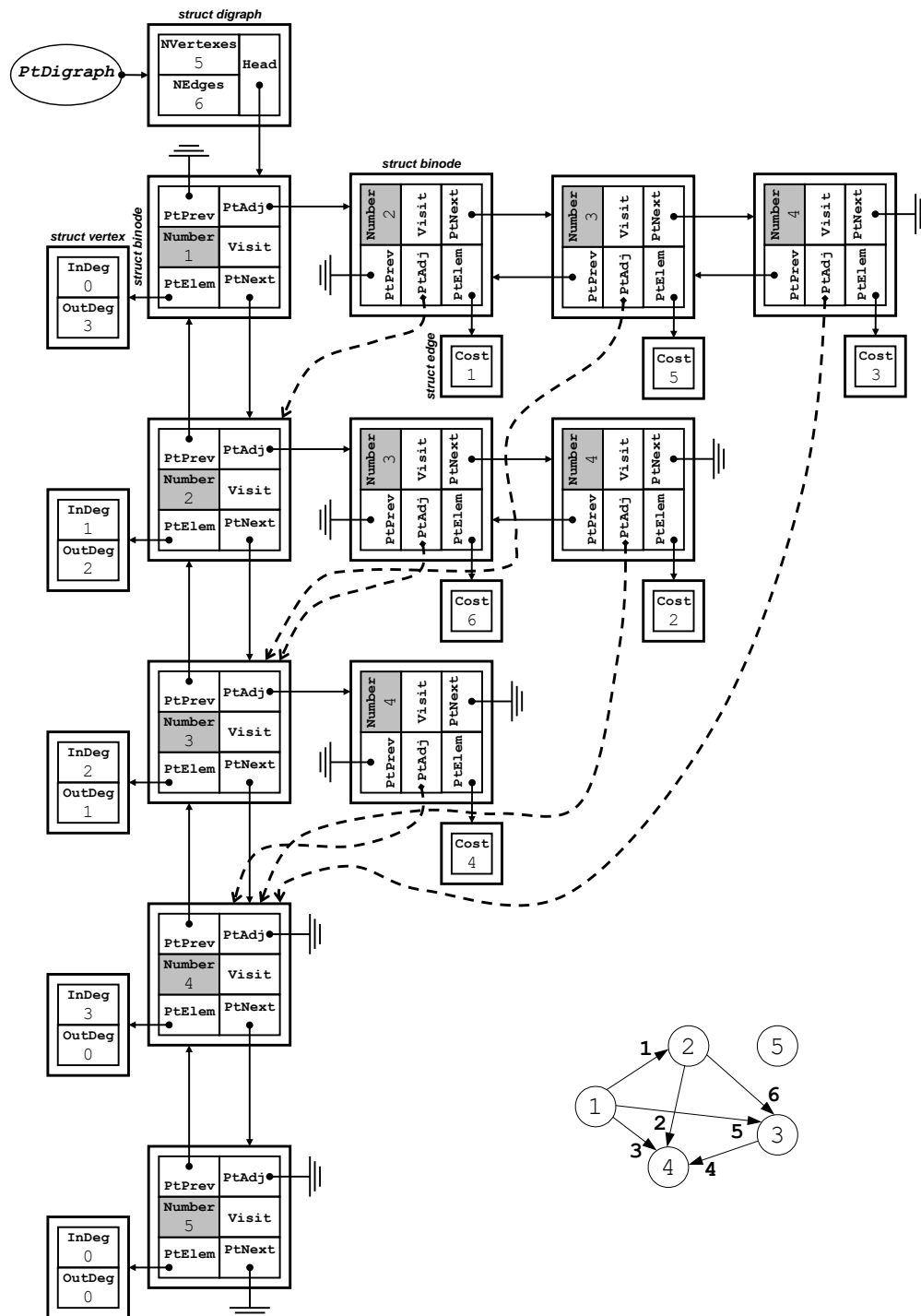


AULAS 12 E 13 - DÍGRAFOS

Comece por ler o Capítulo 10 – Grafos (ver 10.3 – Implementação do Grafo, 10.4 – Caracterização do Grafo e 10.5 – Dígrafo/Grafo dinâmico, páginas 456-470). Estude o funcionamento da implementação do dígrafo/grafos dinâmico e analise a sua implementação, para se familiarizar com os algoritmos.

O tipo de dados abstrato DIGRAPH_DYNAMIC é constituído pelo ficheiro de interface **digraph.h** e pelo ficheiro de implementação **digraph.c** e implementa a manipulação de dígrafos/grafos dinâmicos, usando listas biligadas genéricas para manter as listas dos vértices e das arestas ordenadas por ordem crescente, tal como se mostra na figura seguinte.



Para testar o tipo de dados – as operações básicas sobre dígrafos e as operações propostas para a aula e projeto final – é fornecido o programa de simulação gráfica **simdigraph.c** e a *makefile* **mkdigraph**. Comece por testar convenientemente toda a sua funcionalidade básica.

Depois, acrescente-lhe a seguinte funcionalidade:

- Verificar de que tipo é um vértice. A função deve ter o seguinte protótipo:

```
int VertexType (PtDigraph pdig, unsigned int pv);
```

A função deve devolver: NO_DIGRAPH (se o dígrafo não existir); DIGRAPH_EMPTY (se o dígrafo estiver vazio); NO_VERTEX (se o vértice não existir); SINK se ele for um vértice sumidouro; SOURCE se ele for um vértice fonte; DISC se ele for um vértice desconexo; ou OK se for um vértice comum.

Teste a função para o **digrafo5.txt**, que é constituído por 5 vértices e 6 arestas, sendo que o vértice 1 é um vértice fonte, o vértice 4 é um vértice sumidouro e o vértice 5 é um vértice desconexo.

Adicionar ao tipo de dados abstrato funções que permitam efetuar as seguintes operações:

- Determinar a centralidade de um dado vértice. A centralidade é um valor real entre 0.0, quando o vértice não tem vértices sucessores, e 1.0 quando esse vértice tem todos os outros vértices do dígrafo como sucessores. A função deve ter o seguinte protótipo:

```
int VertexOutDegreeCentrality (PtDigraph pdig, unsigned int pv, double *pcent);
```

A função atribui a pcent a centralidade do vértice. E devolve os seguintes valores de retorno: OK, NO_DIGRAPH, DIGRAPH_EMPTY, NO_VERTEX ou NULL_PTR (se o ponteiro pcent for NULL).

- Determinar o (primeiro) vértice do dígrafo que tenha a máxima centralidade. A função deve ter o seguinte protótipo:

```
int MaxOutDegreeCentrality (PtDigraph pdig, unsigned int *pv, double *pmax);
```

A função atribui a pv a identificação do vértice e a pmax a sua centralidade. E devolve os seguintes valores de retorno: OK, NO_DIGRAPH, DIGRAPH_EMPTY ou NULL_PTR (se algum dos ponteiros pv e pmax for NULL).

- Determinar a média do número de sucessores (Out-Degree) dos vértices sucessores de um dado vértice. A função deve ter o seguinte protótipo:

```
int AverageSucessorOutDegree (PtDigraph pdig, unsigned int pv, double *pmsuc);
```

A função atribui a pmsuc a média do número de sucessores dos vértices sucessores do vértice. E devolve os seguintes valores de retorno: OK, NO_DIGRAPH, DIGRAPH_EMPTY, NO_VERTEX, NO_EDGE (se o vértice não tiver vértices sucessores) ou NULL_PTR.

- Criar uma fila contendo os vértices desconexos de um dado dígrafo. A função deve ter o seguinte protótipo:

```
int AllIsolates (PtDigraph pdig, PtQueue *pqueue);
```

A função cria uma fila de inteiros contendo a identificação dos vértices desconexos e devolve a referência da fila em pqueue ou NULL caso não haja memória para criar e preencher a fila. E devolve os seguintes valores de retorno: OK, NO_DIGRAPH, DIGRAPH_EMPTY, NULL_PTR (se o ponteiro pqueue for NULL) ou NO_MEM (caso não seja possível criar ou inserir elementos na fila).

- Criar uma fila contendo os vértices predecessores de um dado vértice. A função deve ter o seguinte protótipo:

```
int AllPredecessors (PtDigraph pdig, unsigned int pv, PtQueue *pqueue);
```

A função cria uma fila de inteiros contendo a identificação dos vértices predecessores do vértice pv e devolve a referência da fila em pqueue ou NULL caso não haja memória para criar e preencher a fila. E devolve os seguintes valores de retorno: OK, NO_DIGRAPH, DIGRAPH_EMPTY, NO_VERTEX, NULL_PTR ou NO_MEM.

- Criar uma fila contendo os vértices que definem as arestas inexistentes no dígrafo. A função deve ter o seguinte protótipo:

```
int AllNonEdges (PtDigraph pdig, PtQueue *pqueue);
```

A função cria uma fila de inteiros com os pares de vértices (vértice emergente seguido do vértice incidente) que definem as arestas inexistentes no dígrafo e devolve a referência da fila em pqueue ou NULL caso não haja memória para criar e preencher a fila. A função devolve os seguintes valores de retorno: OK, NO_DIGRAPH, DIGRAPH_EMPTY, NULL_PTR ou NO_MEM.

Atenção:

Apesar de habitualmente considerarmos que os vértices se encontram sequencialmente numerados, com início em 1, deve implementar os algoritmos de maneira o mais versátil possível. Ou seja, deve sempre varrer e processar a lista de vértices do dígrafo.

Também deve respeitar os protótipos das funções propostos para poder simular toda a funcionalidade com o programa **simdigraph.c**.