

Algoritmos e Complexidade

2013/2014 – 2º Semestre

1º Trabalho – O Tipo de Dados Abstrato Dígrafo

– Data limite de entrega: 11 de Junho –

Objectivo

O tipo de dados abstrato Dígrafo/Grafo permite representar e efectuar diversas operações sobre dígrafos e grafos. Pretende-se terminar o desenvolvimento do tipo de dados abstrato iniciado nas aulas práticas e acrescentar-lhe algumas funcionalidades adicionais. Para isso, e com base nos ficheiros **digraph.h** e **digraph.c** das aulas práticas, deve:

1 – Concluir a implementação das funções do tipo de dados abstrato Dígrafo/Grafo indicadas no guião das aulas práticas.

2 – Adicionar ao tipo de dados abstrato Dígrafo/Grafo funções que permitam efectuar as seguintes operações:

- a. Verificar se um dado **dígrafo** G é **completo**. A função deve ter o seguinte protótipo:

```
int DigraphComplete (PtDigraph pdig, unsigned int *pcomp);
```

A função atribui a `pcomp` o valor 1, se o dígrafo for completo, e o valor 0, caso contrário. E devolve os seguintes valores de retorno: OK, NO_DIGRAPH (se o dígrafo não existir), DIGRAPH_EMPTY (se o dígrafo estiver vazio) ou NULL_PTR (se o ponteiro for NULL).

- b. Construir o **dígrafo transposto** G^T de um dado dígrafo G . O dígrafo transposto G^T é um dígrafo com os mesmos vértices e com as mesmas arestas do dígrafo G , estando as arestas invertidas. A função deve ter o seguinte protótipo:

```
PtDigraph DigraphTranspose (PtDigraph pdig);
```

A função começa por criar um dígrafo nulo e de seguida insere os vértices. Depois, insere as arestas simétricas e, finalmente, devolve a referência do dígrafo criado ou NULL, no caso de inexistência de memória.

- c. Verificar se um dado dígrafo G é **fortemente conexo**, isto é, se existe um caminho entre qualquer par de vértices do dígrafo G . A função deve ter o seguinte protótipo:

```
int DigraphStronglyConnected (PtDigraph pdig, unsigned int *pstrong);
```

A função atribui a `pstrong` o valor 1, se o dígrafo for fortemente conexo, e o valor 0, caso contrário. E devolve os seguintes valores de retorno: OK, NO_DIGRAPH, DIGRAPH_EMPTY, NO_MEM (se não existir memória para criar as sequências necessárias para invocar o algoritmo de Dijkstra) ou NULL_PTR.

- d. Implementar o **fecho transitivo** de um dado dígrafo G. Uma aresta (v_i, v_j) é inserida no dígrafo, se e só se, v_j é alcançável a partir de v_i e essa aresta ainda não existe no dígrafo. Considere que as novas arestas a inserir no dígrafo têm custo unitário. A função deve ter o seguinte protótipo:

```
int DigraphTransitiveClosure (PtDigraph pdig);
```

A função devolve os seguintes valores de retorno: OK, NO_DIGRAPH, DIGRAPH_EMPTY 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.

Para construir o fecho transitivo e verificar se um dígrafo é fortemente conexo deverá obrigatoriamente utilizar a função que determina os vértices alcançáveis (função **Reach**) que foi proposta no guião das aulas práticas.

Também deve respeitar os protótipos das funções propostos para poder simular toda a funcionalidade (com excepção da função **DigraphComplete**) com o programa **simdigraph.c**.

Note que para o correcto funcionamento e teste do tipo de dados abstrato Dígrafo/Grafo também é necessário desenvolver o tipo de dados abstrato Fila com Prioridade de Dijkstra (aula prática nº11).

Bibliografia aconselhada

António Adrego da Rocha, Estruturas de Dados e Algoritmos em C, 3ª Edição Revista e Aumentada, FCA – Editora de Informática, 2014.

Entrega do trabalho

Devem entregar apenas as implementação dos tipos de dados (**pqueue_dijkstra.c** e **digraph.c**) por correio electrónico para adrego@ua.pt. Os dois ficheiros devem ser enviados num ficheiro *zip* com o seguinte nome **trab1_nmecanográficoaluno1_nmecanográficoaluno2**.

Atenção: os alunos repetentes ou que não tiveram aproveitamento a Programação II devem realizar o trabalho individualmente. Os restantes alunos podem realizar o trabalho em grupo.