



## Laboratório 5 - Acervos

Semana de 16 a 20 de Novembro de 2015

Duração: 2 horas

O objectivo desta aula é o estudo e manipulação de acervos (*heaps*). Os ficheiros `heap.c` e `heap.h` contêm a implementação e a interface para um tipo `heap`. Este tipo de dados está implementado como um tipo abstracto o que lhe confere grande versatilidade. Nesta implementação, o acervo contém elementos do tipo `Item`, definido no ficheiro `Item.h` que é igualmente incluído. Alterando o tipo de dados `Item`, o mesmo código pode ser usado como acervo de outros tipos de dados.

O ficheiro `client.c` contém um programa que permite experimentar várias operações sobre acervos contendo números inteiros (até 16 números; para acervos maiores terá de alterar a definição em `client.c`). Utilize a opção "h" do menu do `client.c` para verificar as operações disponíveis ou a desenvolver.

Examine cuidadosamente o código contido nos ficheiros `heap.h` e `heap.c`.

1. Compile e execute o programa cliente. Estude o exemplo (opção "e" do menu do `client.c`). Qual é a condição de acervo que está a ser utilizada? Verifique no código como é testada a condição de acervo.
2. Insira vários números inteiros no acervo. Verifique o funcionamento da rotina `FixUp()`. A ordem própria do acervo é mantida?
3. Altere o valor de um elemento do acervo, aumentando e diminuindo sucessivamente o valor. A ordem é mantida no acervo? Quais são as funções usadas para o efeito?
4. Remova várias vezes o maior elemento do acervo. O que é que acontece? Quais são as funções necessárias para obter este resultado?
5. Complete a função `CleanHeap()`, que apaga todos os elementos de um acervo existente libertando a memória alocada. Repare que esta funcionalidade está associada à opção "c" do menu do `client.c`. Assegure-se ainda que na saída do programa toda a memória alocada é libertada.
6. Complete a função `VerifyHeap()`, que verifica se uma tabela representa um acervo, ou seja, satisfaz a condição de acervo. Verifique que esta funcionalidade está associada à opção "v" do menu do `client.c`.
7. Complete a função `Heapify()`, que transforma uma tabela num acervo. A função a desenvolver recebe como argumento uma estrutura `Heap`, na qual não é garantido que se verifique a condição de acervo. A função deve efectuar as trocas necessárias para garantir que a estrutura representa um acervo. Repare que esta funcionalidade está associada à opção "f" do menu do `client.c`.
8. Complete a função `HeapSort()`, que ordena por ordem crescente os elementos do acervo (algoritmo `heapsort`). Repare que esta funcionalidade está associada à opção "s" do menu do `client.c`. Qual é a complexidade do algoritmo de ordenação?