

AULA 10 – TIPOS ABSTRATOS DE DADOS

***** Entregue, num ficheiro ZIP, os 2 ficheiros .cpp deste guião *****

Pretende-se desenvolver o tipo abstrato de dados **SCHEDULING SEQUENCE**, que permitirá representar uma sequência de intervalos de tempo não sobreponíveis, constituindo um conjunto ordenado de tarefas que definem uma sequência de escalonamento.

Mas, neste guião, esse conjunto ordenado será representado usando uma árvore binária de procura (“Binary Search Tree” – BSTree) de intervalos de tempo.

TAREFAS

- Começar por analisar o conteúdo dos ficheiros disponibilizados e as funcionalidades definidas para cada tipo de dados.
- **BSTREE** – Analisar, completar e testar o tipo de dados.
- **SCHEDULINGSEQUENCE** – Analisar, completar e testar esta **nova versão** do tipo de dados.
- **Nota:** os ficheiros surgem com a extensão .cpp (e não .c) por compatibilidade com o **CodeCheck**; mas é usada a linguagem C.
- Teste de modo incremental todo o código desenvolvido: **CodeCheck** ou **Makefile**.
- Pode executar o **valgrind** para verificar se tem “*memory leaks*” ou outros problemas relacionados com a alocação dinâmica de memória.

BINARY SEARCH TREE

- Analisar as **funcionalidades** disponibilizadas e a **estrutura de dados** que permite representar uma árvore binária de procura.
- Responder às **questões** e completar as **funções incompletas** (assinaladas com reticências) do ficheiro **BSTree.cpp**.
- Analisar o ficheiro **BSTreeTest.cpp** e testar as funções terminadas. Que **instâncias** são armazenadas na árvore binária? Como são feitas as **comparações** entre essas instâncias?

SCHEDULING SEQUENCE

- Analisar as **funcionalidades** disponibilizadas e o modo **como é representada uma sequência de intervalos de tempo**.
- Terminar o desenvolvimento do tipo de dados: **funções incompletas** (assinaladas com reticências) no ficheiro **SchedulingSequence.cpp**.
- Analisar o ficheiro **TAD4Test.cpp** e testar as funções terminadas.