Relatório Projeto 3.4 AED 2020/2021 Versão 1.0

Nome: Miguel Sá Pedroso Nº Estudante: 2019218176

TP (inscrição): 5 *Login* no *Mooshak*: AED2019218176

Nº de horas de trabalho: 3 *H* *Aulas Práticas de Laboratório:* 2 *H* *Fora de Sala de Aula:* 1 *H*

(A Preencher pelo Docente) CLASSIFICAÇÃO:

Comentários:

Estrutura de Dados Principal usada em cada sub-projeto:

PROJ 3.1 Perfect binary tree

PROJ 3.2 AVL tree

PROJ 3.3 Splay tree

Estruturas de Dados usadas	Perfect binary tree	AVL tree	Splay tree
VANTAGENS GERAIS (max 3)	• Acessa aos dados em O(log n) •Pesquisas mais rápidas que numa lista ligada •	• Acessa aos dados em O(log n) • Mais rápida que uma BST ou uma Red-Black Tree • Capacidade de se auto-equilibrar	• Rápido acesso aos nós mais utilizados • As operações de inserção, consulta e remoção são realizadas em tempo amortizado O(log N) •
DESVANTAGENS GERAIS (max 3)	•Inserções mais lentas do que numa lista ligada •Implementação mais complexa do que uma lista ligada •	•Inserção de nós lenta • Mais difícil de implementar que uma BST • Remoção de nós da árvore implica bastantes rotações	• Possibilidade de degenerar e ter complexidade O(n), devido a não se equilibrar sozinha • Mais difícil de implementar que uma BST • Necessidade de mais operações na inserção e consulta de dados do que numa BST
Justificação para a escolha no PROJ 3.1	Foi implementada uma árvore binária, uma vez que o acesso aos dados é mais rápido do que numa lista ligada. Como se tratava de uma merkle tree, era necessário aceder aos dados já inseridos na estrutura para poder inserir os hash codes que faltavam, levando a um elevado número de acessos aos dados .		
Justificação para a escolha no PROJ 3.2	Necessidade de uma estrutura de dados onde a quantidade de consultas aos dados é muito superior à sua inserção. Como a árvore AVL se auto-equilibra ao inserir os dados, a operação se inserção é mais demorada do que a consulta. A consulta, devido ao equilíbrio da árvore, é sempre feita em O(log n).		
Justificação para a escolha no PROJ 3.3	Necessidade de uma estrutura de dados que permitisse rápido acesso a um pequeno conjunto de dados acedidos com grande frequência, o que é permitido pela Splay tree, que guarda junta à raiz todos os dados acedidos frequentemente.		