

Relatório Projeto 3.4 AED 2020/2021 Versão 1.0

Nome: Tomás Batista Mendes N° Estudante: 2019232272

TP (inscrição): PI 2 Login no Mooshak: 2019232272

N° 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 Merkel tree (full binary tree)

PROJ 3.2 AVL tree

PROJ 3.3 Splay tree

Estruturas de Dados usadas	<u>Full binary tree</u>	<u>AVL tree</u>	<u>Splay tree</u>
VANTAGENS GERAIS (max 3)	<ul style="list-style-type: none"> Complexidade das operações é $O(\log n)$ 	<ul style="list-style-type: none"> Complexidade das operações é $O(\log n)$ 	<ul style="list-style-type: none"> Acesso rápido aos elementos acessados recentemente
DESVANTAGENS GERAIS (max 3)	<ul style="list-style-type: none"> Consulta pode ser $O(N)$, se todos os nós forem consultados 	<ul style="list-style-type: none"> Complexidade da inserção de um novo nó é $O(\log n)$ 	<ul style="list-style-type: none"> Tempos de acesso alto para os nós não recentemente acessados
Justificação para a escolha no PROJ 3.1	<p>Para este projeto, foi implementada uma Merkel tree, que é basicamente uma árvore binária onde as folhas contêm os elementos da tabela hash.</p>		
Justificação para a escolha no PROJ 3.2	<p>Para este projeto, como o objetivo é consultar muito mais utentes do que inseri-los, uma AVL tree é o ideal. Como a árvore AVL mantém o equilíbrio, a complexidade das operações é $O(\log n)$.</p>		
Justificação para a escolha no PROJ 3.3	<p>Para este projeto, como o objetivo era apenas consultar, com bastante frequência, apenas alguns elementos, uma splay tree é a melhor opção, pois os elementos frequentemente acessados são movidos para a raiz, reduzindo o tempo de acesso.</p>		