Relatório Projeto 3.4 AED 2020/2021 Versão 1.0

Nome: António Marques Maria Nº Estudante: 2017265346

TP (inscrição): 4 *Login* no *Mooshak*: 2017265346

Nº de horas de trabalho: 5 *H Aulas Práticas de Laboratório: 3 H Fora de Sala de Aula: 2 H*

**(A Preencher pelo Docente) CLASSIFICAÇÃO: Comentários:**

# Estrutura de Dados Principal usada em cada sub-projeto:

PROJ 3.1 Árvore Merkle

PROJ 3.2 Árvore AVL

PROJ 3.3 Árvore Splay

|  |  |  |  |
| --- | --- | --- | --- |
| Estruturas de  Dados usadas | Árvore Merkle | Árvore AVL | Árvore Splay |
| VANTAGENS GERAIS (max 3) |  Complexidade constante para retornar a raíz   Pode ser representada por array, permitindo armazenamento eficaz.   São imutáveis, alterar uma transação altera a árvore inteira, permitindo integridade dos dados. |  Pesquisa eficiente em todos os elementos   Está sempre balanceada permitindo sempre tempo logarítmico nas operações.   |  Pesquisa eficiente se os dados tiverem sido recentemente inseridos ou pesquisados   Não precisa de armazenar informação extra como a cor do nó ou a sua altura.   Fácil de implementar |
| DESVANTAGENS GERAIS (max 3) |  É preciso percorrer a árvore inteira para calcular a raiz   Não permite pesquisar um elemento de forma eficiente   |  As inserções são mais complexas devidos às rotações constantes   É mais complicada de implementar do que alternativas como BST.   |  Não é balanceada como a AVL ou VP, podendo acabar degenerada   Envolve várias rotações em cada operação de consulta e inserção.   |
| Justificação para a escolha no PROJ 3.1 | Como a árvore tinha que ser construída bottom-up, escolhi representar a árvore  através de um array, uma vez que era uma árvore binária perfeita isso é  possível, e preencher do final do array até ao início, em vez de alocar todos os nós da árvore e preenchê-los da base até à raíz. | | |
| Justificação para a escolha no PROJ 3.2 | Como as consultas eram muito mais frequentes que as inserções, uma árvore  AVL é mais eficiente uma vez que poupa tempo na pesquisa de um elemento,  compensando o facto da inserção demorar mais tempo devido às rotações constantes. | | |
| Justificação para a escolha no PROJ 3.3 | A árvore splay mantém os elementos mais recentemente acedidos perto da raiz,    por isso se as consultas estiverem concentradas num número reduzido de elementos,  estes vão estar sempre próximos da raiz e o tempo de consulta será reduzido. | | |