

# Relatório Projeto 3.1 AED 2020/2021 Versão 1.0

Nome: João Carlos Borges Silva

Nº Estudante: 2019216753

TP (inscrição): PL8 Login no Mooshak: 2019216753

Nº de horas de trabalho: 20 Min Aulas Práticas de Laboratório: 0 H Fora de Sala de Aula: 20 Min

(A Preencher pelo Docente) CLASSIFICAÇÃO:

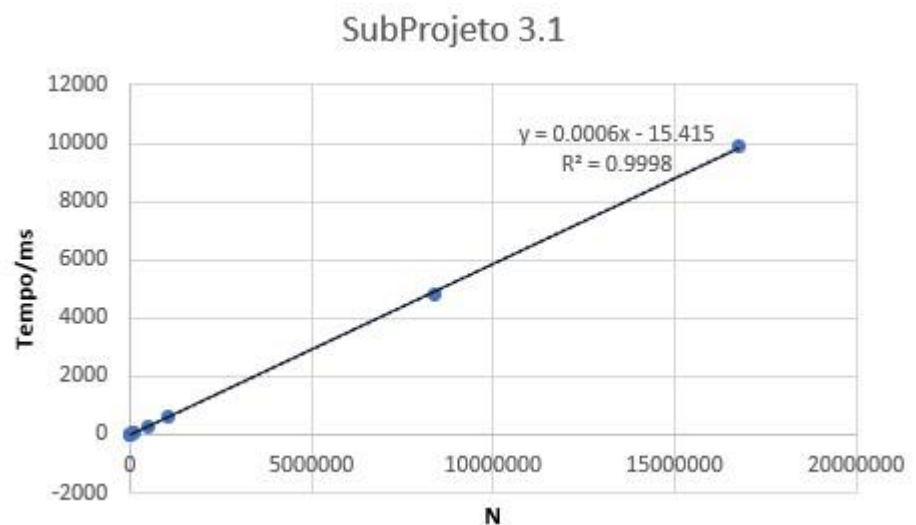
Comentários:

## 1. Análise Empírica de Complexidade

Correr a implementação do projeto 3.1 para um número crescente de transações e obter os tempos de execução (excluindo tempo de leitura e impressão de resultados). Produzir tabela, gráfico e regressão relevantes.

$$f(N) = 0.006N - 15.415$$

N	tempo/ms
1024	0.99
8192	4.344
16384	9.328
65536	35.348
131072	70.1
524288	297.713
1048576	592.292
8388608	4779.55
16777216	9897.273



A expressão  $f(N)$  está de acordo com o esperado? Justifique.

A expressão obtida estará de acordo com o esperado. Como podemos perceber o número de operações sobre as transações será sempre  $2N-1$  (sendo  $N$  o número de elementos iniciais), isto porque  $N$  será sempre um número de base 2. Ou seja neste caso apresentaria uma complexidade digamos linear ( $O(n)$ ), logo é natural que a expressão  $f(N)$  seja de facto uma reta afim.

O projeto 3.1 pode ser implementado seguindo uma abordagem iterativa e uma recursiva.

Explique sucintamente o essencial das duas implementações em termos de estruturas de dados utilizadas e da propagação dos *hashcodes* na árvore.

Numa abordagem iterativa teríamos um algoritmo com 2 ciclos (“for” ou “while”, tipo de implementação escolhida) onde a árvore propagar-se-ia da base para a raiz, onde a estrutura poderia ser um array. Já numa abordagem recursiva teríamos uma chamada constante da própria função com propagação desde a raiz até à base da árvore, onde neste caso iríamos usar uma árvore como estrutura de dados. Ainda em ambos os casos, a sua complexidade e o cálculo dos hashcodes(baixo para cima) manter-se-ia igual para ambas as abordagens.