Relatório Projeto 4.2 AED 2021/2022

Nome: Sancho Amaral Simões

Nº Estudante: 2019217590

PL (inscrição): PL2 *Login* no *Mooshak:* SanchoAmaralSimoes

**Tabela (S3) Gráfico (S1)**

|  |  |
| --- | --- |
| N | Tempo(s) |
| 100000 | 0,1232 |
| 200000 | 0,4321 |
| 300000 | 0,7326 |
| 400000 | 1,2230 |
| 500000 | 1,7321 |
| 600000 | 2,4521 |
| 700000 | 2,8287 |
| 800000 | 3,3293 |
| 900000 | 3,9412 |
| 1000000 | 4,7421 |

(1) Descreva sucintamente as otimizações feitas ao *QuickSort*. A expressão *O(f(n))* está de acordo com o esperado? Justifique.

As otimizações efetuadas no *quicksort* são ao nível da escolha do *pivot*: de maneira a equilibrar o tamanho de ambas os *subarrays* a ordenar pelas duas chamadas recursivas, é escolhido um *pivot* correspondente à mediana do *array* original. Deste modo, minimiza-se ao máximo o número de chamadas recursivas e, portanto, o número de operações de *push*/*pop* na stack o que origina uma considerável melhoria na eficiência temporal do algoritmo. A expressão *Big-O* prevista para a eficiência temporal do algoritmo é de *O(N log(N))*, onde *N* é o tamanho do *input* (*raster*). Este facto é comprovado pela proximidade dovalor *R2*de 1 patente no gráfico acima.

(2) Qual a expressão O(f(n)) para a complexidade espacial na solução *S3*? Justifique.

A expressão *O(f(n))* para a complexidade espacial na solução *S3* corresponde à complexidade espacial do *quicksort*, que é de *O(N)*, onde *N* é o tamanho do *input* (*raster*), visto que o *quicksort* é um algoritmo que aplica o ordenamento *in situ*, isto é, no próprio *array*, sem recorrer à criação de *arrays* auxiliares, como por exemplo acontece no *merge sort*.