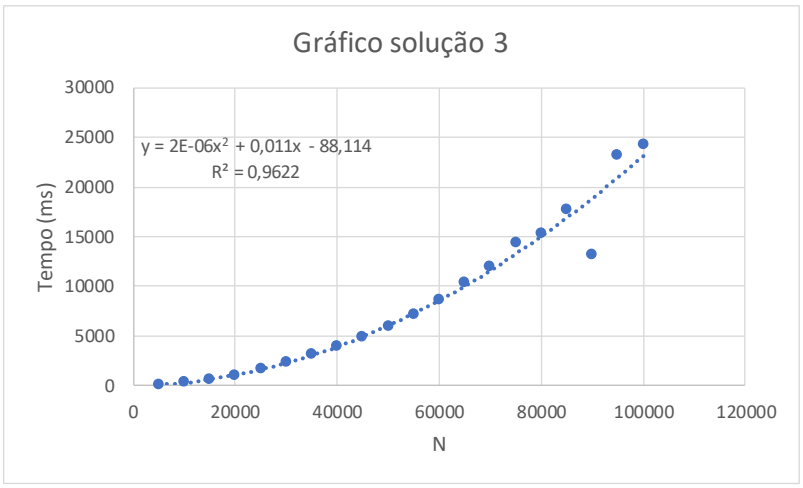
Relatório Projeto 4.2 AED 2021/2022

Nome: João Emanuel Sousa Moreira
Nº Estudante: 2020230563
PL (inscrição): PL1
Login no *Mooshak:* 2020230563

Tabela (S3)

N	Times (ms)
5000	60
10000	239
15000	535
20000	938
25000	1558
30000	2245
35000	3064
40000	3866
45000	4856
50000	5854
55000	7158
60000	8540
65000	10285
70000	11851
75000	14266
80000	15293
85000	17712
90000	13126
95000	23181
100000	24191

Gráfico (S3)



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

A má escolha do pivot pode fazer a complexidade variar entre O (n log n) e O (n^2). Escolhemos para pivot a mediana dos elementos: do início, do meio e do fim. Estes são previamente ordenados. No fim da escolha, o pivot é colocado no inicio do *array*.

Quando encontrado um elemento igual ao pivot fazemos *swap*. Isto permite que a repartição do *array* seja mais bem distribuída.

Quando temos menos de 30 elementos no array, usamos o *insertion sort*. Este algoritmo funciona bem para poucos elementos e semelhantes.

Poupamos um pouco em que o ponteiro do fim, esta a apontar para o penúltimo elemento, visto que o ultimo será maior do que o pivot.

A curva obtida está de acordo com o esperado porque assemelha-se a O (n log n).

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

A complexidade espacial deste algoritmo é no melhor caso O (log n) e O (n) no pior caso.
Por cada chamada recursiva, nos guardamos 3 ponteiros (inicio, fim e pivot), O (1). Visto que fazemos log n chamadas recursivas a complexidade resulta em O (log n).
O pior caso, seria ter um *array* sequencialmente/inversamente ordenado, onde iriamos fazer n chamadas recursivas.