

Relatório Projeto 4.3 AED 2021/2022

Nome: Mariana Lopes Paulino

Nº Estudante: 2020190448

TP (inscrição): 1

Login no Mooshak: 2020190448

Registrar os tempos computacionais do Quicksort para os 3 métodos de seleção do pivot e para os diferentes tipos de sequências. Os tamanhos das sequências (N) devem ser: 25000, 50000, 75000, 100000, 125000. Só deve ser contabilizado o tempo de ordenamento. Exclui-se o tempo de leitura do input e de impressão dos resultados. Devem apresentar e discutir as regressões para cada variante e para cada tipo de sequência.

Gráfico para SEQ_ALEATORIA

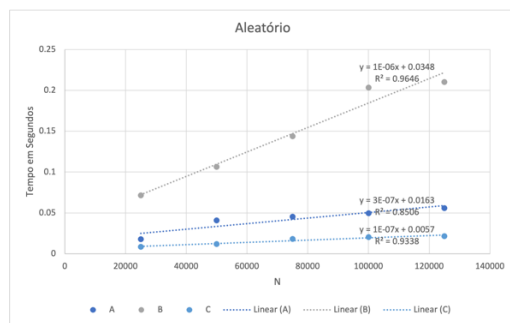


Gráfico para SEQ_ORDENADA_DECRESCENTE

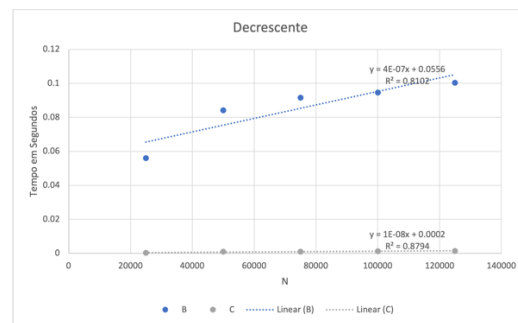
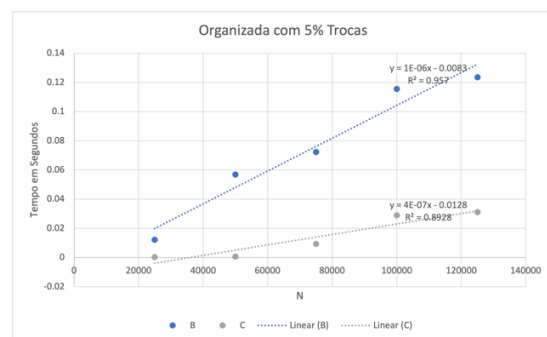


Gráfico para SEQ_QUASE_ORDENADA_1%



Gráfico para SEQ_QUASE_ORDENADA_5%



Análise dos resultados:

O QuickSort é um algoritmo de ordenação baseado em recursividade e sucessivas divisões do array que necessita de ser ordenado com o fim da sua ordenação fixando um pivot neste caso existiam 3 métodos: A onde a escolha do pivot é fixada no primeiro índice do array, B onde a escolha do pivot é aleatória dentro do array, isto é, qualquer um dos elementos podem ser pivot e C onde o pivot fixado é o resultado do cálculo da mediana de 3 valores (inicial, meio, final). A partir da fixação do pivot separamos os restantes elementos em dois arrays, um com os elementos menores que este à sua esquerda e à sua direita ficam os elementos que sejam maiores que o considerado. Vamos partindo sempre o array até apenas termos os elementos todos separados onde os vamos juntando e agrupando até resultar no array ordenado. A complexidade do algoritmo QuickSort no melhor caso pode ser $O(n \cdot \log n)$ quando bem implementado e quando a escolha do pivot é bem feita. Caso isto não aconteça a sua complexidade pode rapidamente passar a $O(n^2)$. Com as três escolhas diferentes de pivot é necessária uma análise mais cuidada em particular. Ao correr os tempos do método A este

apenas nos vai dar resultados apenas na sequência aleatória dando nas outras (sequência ordenada decrescente e sequências quase ordenadas com 1% e 5% de trocas de elementos) erro de recursão (Recursion Error) o que significa que o limite máximo de recursões foi excedido devido tanto à escolha de pivot como às iterações a fazer para que o array fique ordenado, no caso da sequência ser decrescente o pivot vai ser sempre o menor pelo que para separar e juntar todos os elementos vai ter de fazer n iterações para os separar todos e n iterações para os voltar a juntar tendo o pivot como primeiro elemento de uma sequência fazendo com que este caso seja um dos piores casos, o mesmo acontece para as sequências quase ordenadas. Nos gráficos obtidos conseguimos observar que todas as linhas representadas seguem uma função logarítmica, conseguindo concluir assim que seguem a complexidade $O(n \cdot \log n)$ obtendo assim uma boa implementação do QuickSort uma vez que a sua má implementação iria aumentar a sua complexidade até $O(n^2)$.