

# Relatório Projeto 4.3 AED 2020/2021

Nome: Tomás Batista Mendes

Nº Estudante: 2019232272

TP (inscrição): 2 Login no Mooshak:

Nº de horas de trabalho: 7 H Aulas Práticas de Laboratório: 4 H Fora de Sala de Aula: 3 H

(A Preencher pelo Docente) CLASSIFICAÇÃO:

Comentários:

Registrar os tempos computacionais do RS para os diferentes tipos de sequências. O tamanho das sequências (N) deve ser crescente e terminar em 10,000,000. 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 tipo de sequência.

Gráfico para SEQ\_ALEATORIA

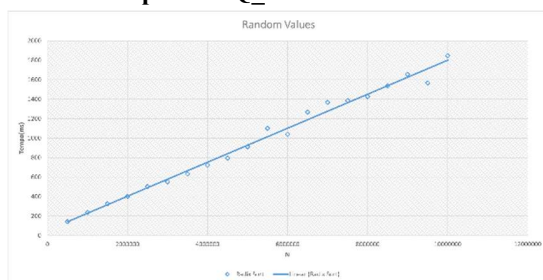


Gráfico para SEQ\_ORDENADA\_DECRESCENTE

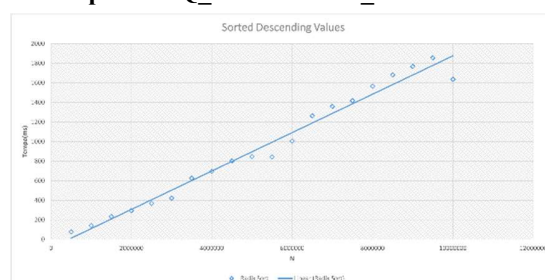


Gráfico para SEQ\_QUASE\_ORDENADA\_1%

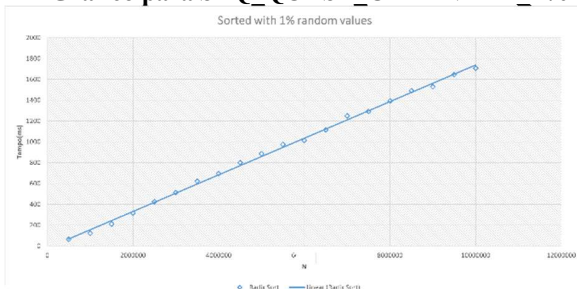
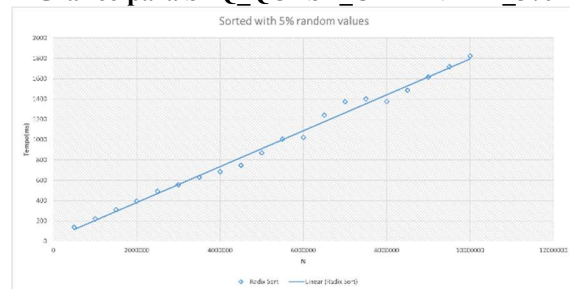


Gráfico para SEQ\_QUASE\_ORDENADA\_5%



**Análise dos resultados, discutindo a implementação alternativa do RS (MSD ou LSD) e considerando também a complexidade espacial do algoritmo:**

Para este projeto foi usada a versão LSD do Radix Sort visto que a sua implementação é mais simples e não precisa de um grande overhead de memória como o MSD. Para o MSD conseguir dar sort, a sua implementação é ligeiramente diferente, sendo preciso guardar num “balde” diferente todos os caracteres ou numeros anteriores. Se não o fizéssemos, o array final não ficaria sorted. A versão implementada do LSD considera os numeros em base 10, sendo a sua complexidade temporal  $O(N + 10)$ . Em relação á complexidade espacial, o LSD apenas precisa de memória suficiente para guardar uma cópia do array original e dos possíveis valores dos “baldes”, neste caso 0-9 por ser em base 10. Se fosse em base 2 apenas seriam precisos 2 “baldes”, um para os 0's e outro para os 1's, e a complexidade temporal seria  $O(N + 2)$ . Em relação aos tempos de execução, estes estão de acordo com o esperado, demorando basicamente o mesmo tempo para todos os casos, porque o Radix Sort usa o Counting Sort como rotina (pois é um algoritmo estável), que não precisa de fazer comparações para dar sort ao array, resultando assim em tempos muito parecidos para todos os casos.