

# Relatório Projeto 4.1 AED 2020/2021

Nome: Miguel Sá Pedroso

Nº Estudante: 2019218176

TP (inscrição): PL5

Login no Mooshak: AED2019218176

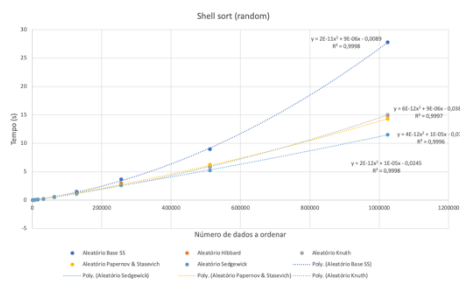
Nº de horas de trabalho: 8 H Aulas Práticas de Laboratório: 2 H Fora de Sala de Aula: 6 H

(A Preencher pelo Docente) CLASSIFICAÇÃO:

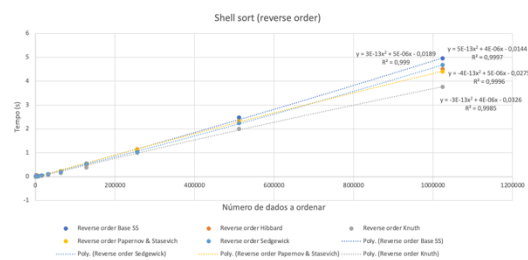
Comentários:

Registrar os tempos computacionais do B-SS e das 4 variantes selecionadas do I-SS 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 a melhor variante em cada tipo de sequência.

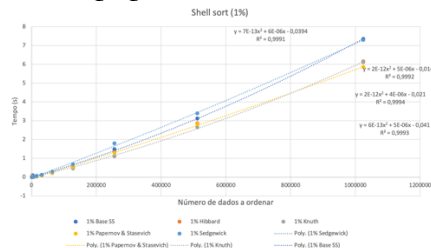
## Gráfico para SEQ\_ALEATORIA



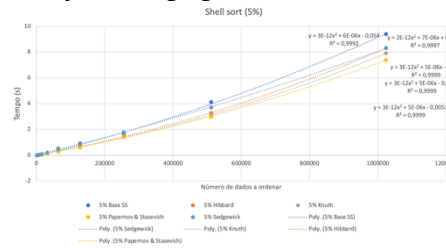
## Gráfico para SEQ\_ORDENADA\_DECRESCENTE



## Gráfico para SEQ\_QUASE ORDENADA 1%



## Gráfico para SEQ\_QUASE ORDENADA 5%



Sequência de incremento ou regra de incremento de cada variante (B-SS, I-SS-1, I-SS-2, I-SS-3, I-SS-4):

B-SS:  $N/2^k$  ( $N/2, N/4, \dots$ )

I-SS-1/Knuth:  $(3^k - 1) / 2$  (1, 4, 13, 40, ...)

I-SS-2/Hibbard:  $2^k - 1$  (1, 3, 7, 15, ...)

I-SS-3/Papernov & Stasevich:  $2^k + 1$  (1, 3, 5, 9, 17...)

I-SS-4/Sedgewick:  $4^i + 1 + 3 \cdot 2^i + 1$  (1, 8, 23, 77, ...)

## Análise dos resultados:

Como se pode observar nos gráficos, todas as versões testadas do Shell Sort demoraram mais tempo a ordenar a sequência aleatória e menos tempo a ordenar a sequência ordenada decrescentemente, sendo que a sequência com 5% de dados desordenados demorou um pouco mais que a sequência com 1%. Em relação às diversas variantes, a versão base foi consistentemente a mais lenta, sendo que as outras apresentaram resultados semelhantes em todos os casos, à exceção do ordenamento dos dados aleatórios, em que a versão com os incrementos de Sedgewick se mostrou particularmente eficiente. Todas as versões apresentaram complexidade  $O(N^2)$  para qualquer caso.