

Relatório Projeto 4.4 AED 2020/2021

Nome: Tomás Batista Mendes

Nº Estudante: 2019232272

TP (inscrição): 2 Login no Mooshak:

Nº de horas de trabalho: 4 H Aulas Práticas de Laboratório: 2 H Fora de Sala de Aula: 2 H

(A Preencher pelo Docente) CLASSIFICAÇÃO:

Comentários:

Registrar os tempos computacionais das variantes em consideração 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.

Gráfico para SEQ_ALEATORIA

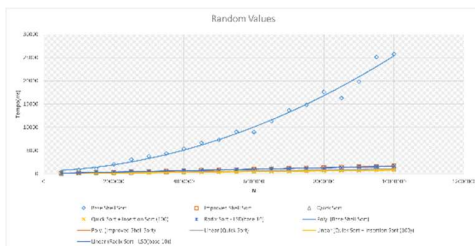


Gráfico para SEQ_ORDENADA DECRESCENTE

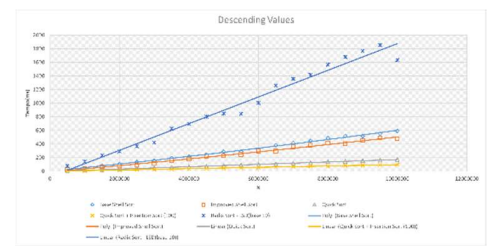
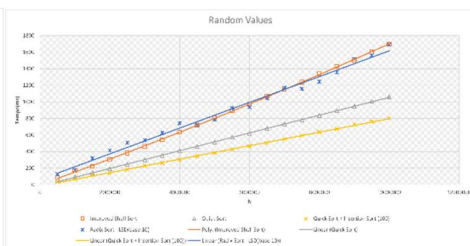


Gráfico para SEQ_QUASE_ORDENADA_1%

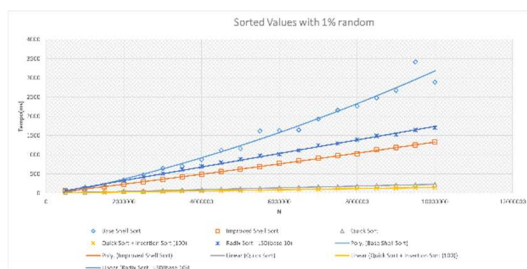
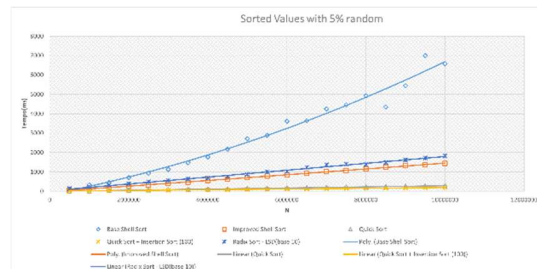


Gráfico para SEQ_QUASE_ORDENADA_5%



Sequência de incremento ou regra de incremento do I-SS para cada tipo de sequência:

Para a versão do I-SS foi usada a sequência de incrementos de 'Tokuda' que se obtém através da fórmula:
$$a(n) = \text{ceiling}((9 * (9/4)^n - 4) / 5)$$

Análise dos resultados considerando também a complexidade espacial dos algoritmos:

Em termos dos tempos de execução e como é possível comprovar pelos gráficos anteriores, o B-SS foi o pior algoritmo em todos os casos exceto no caso da sequência ordenada decrescente porque é o melhor cenário para o SS. Pelo contrário, o QS e a versão melhorada com o IS, demonstraram ser os melhores em todos os casos. De notar que, o RS apesar de ser o que apresenta a melhor complexidade temporal $O(N + k)$, sendo k a base em que está implementado, não obteve melhores resultados por ter sido implementada a versão em base 10. Implementando uma versão numa base mais alta, diminuiria bastante os tempos de execução do algoritmo por ser um algoritmo que não precisa de fazer comparações mas aumentaria bastante a complexidade temporal. Este é também o único algoritmo que tem uma complexidade temporal diferente de $O(N)$, pois todos os outros fazem o *sort in place*, i.e., não precisam de memória adicional para dar *sort*.