## Relatório Projeto 4.4 AED 2020/2021

Nome: Edgar Filipe Ferreira Duarte N° Estudante: 2019216077

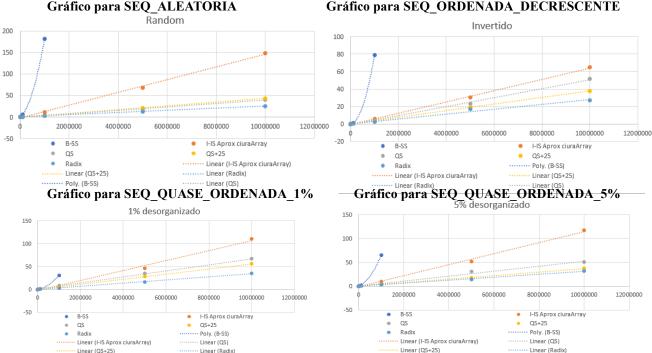
TP (inscrição): PL8 Login no Mooshak: AED2019216077

Nº de horas de trabalho: 2 H Aulas Práticas de Laboratório: 2 H Fora de Sala de Aula: 0 H

(A Preencher pelo Docente) CLASSIFICAÇÃO:

Comentários:

Registar 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.



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

Para todas as sequências foi escolhida uma aproximação da série de Marcin Ciura.

## Análise dos resultados considerando também a complexidade espacial dos algoritmos: (10)

A partir dos resultados obtidos, é notável o ganho de eficiência ao utilizar o *radix*, comparando com os outros métodos de ordenação. Assim, se se pretender um algoritmo em que a velocidade de execução é a prioridade, a utilização de ambas as versões do *ShellSort* não é recomendável, sendo que o *QuickSort*, , para números muito grandes de elementos, também perde bastante eficiência, comparando com o *radix*. A complexidade espacial de cada algoritmo é : O(1) para o *ShellSort*, O(log(n)) , no caso médio, e O(n), no pior caso, para o *QuickSort* e O(n+k) para o *radix*. Assim, se se pretender um algoritmo leve, que corra com fluidez em todos os computadores, para sequências de poucos elementos, pode ser considerado o *ShellSort*, enquanto que, para sequências com número médio de elementos (na ordem do milhão) pode ser considerado o *QuickSort*. Para sequências muito grandes, continua a ser aconselhado a utilização do *radix*.