Relatório Projeto 4.3 AED 2020/2021

Nome: Edgar Filipe Ferreira Duarte Nº Estudante: 2019216077
TP (inscrição): PL 8 *Login* no *Mooshak: AED 2019216077*
Nº de horas de trabalho: 5*H* *Aulas Práticas de Laboratório:* 2 *H* *Fora de Sala de Aula:* 3 *H*

(A Preencher pelo Docente) CLASSIFICAÇÃO:

Comentários:

Registar 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

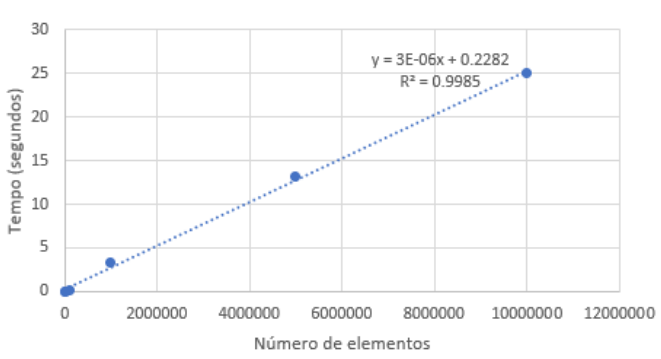


Gráfico para SEQ_ORDENADA_DECRESCENTE

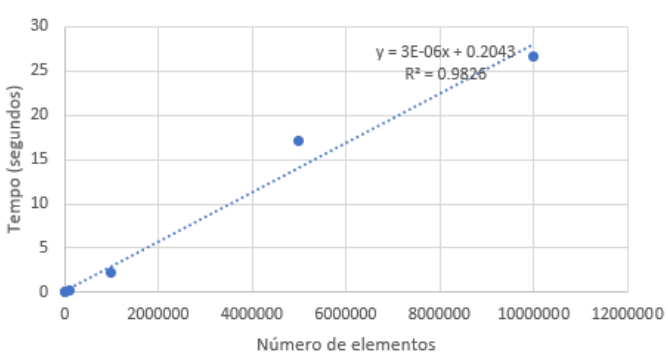


Gráfico para SEQ_QUASE_ORDENADA_1%

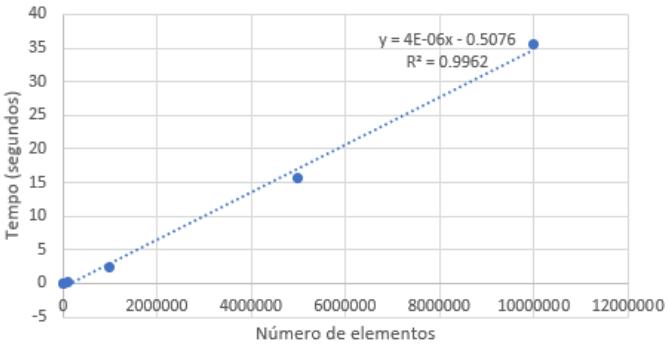
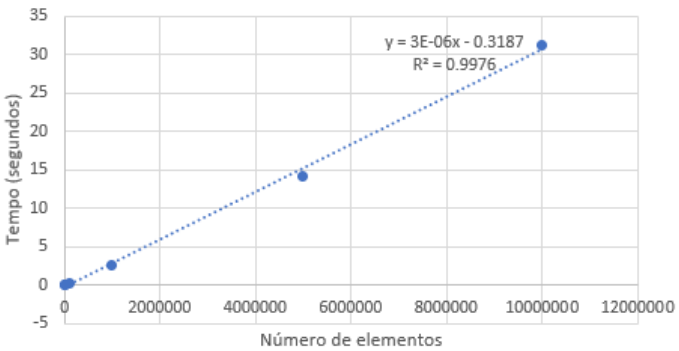


Gráfico para SEQ_QUASE_ORDENADA_5%



Análise dos resultados, discutindo a implementação alternativa do RS (MSD ou LSD) e considerando também a complexidade espacial do algoritmo: (11)

A partir dos dados obtidos, é evidente que, não só, o radix Sort é extremamente eficiente, como também, tem um tempo de execução relativamente constante para todos os cenários. De todos os algoritmos de ordenação até aqui testados, o radix sort é, de longe, o com melhor performance. A implementação abordada foi o LSD, ou seja, a leitura de dígitos da direita para a esquerda. Foi escolhida esta abordagem visto que, embora ela tenha uma menor eficiência em termos de performance do que o MSD, o LSD utiliza menos recursos do que o MSD (memória e complexidade de código). O MSD é mais eficiente do que o LSD visto

que, este percorre a sequência da esquerda para a direita, logo, consegue muito mais rapidamente, quais os números maiores (ex: sequencia 500 400 450, o MSD só precisa de percorrer a sequencia 2 vezes para a ordenar enquanto o LSD precisa de percorrer 3 vezes para descobrir qual a ordem dos números). Em contrapartida, o MSD precisa de guardar em memoria ativa quais os números e posições que já tenham sido ordenadas, gastando muito mais memória. Pelo descrio acima, a complexidade temporal do radix sort é O(n) enquanto que a complexidade espacial é O(n+k).