

Relatório Projeto #1 AED 2024-2025

Nome: *Vasco Guilherme Alves*
PL (inscrição): *PL 6*

Nº Estudante: *2022 22 82 07*

- ⇒ Registrar os tempos computacionais das 3 soluções desenvolvidas.
- ⇒ Os tamanhos dos arrays (N) devem permitir obter dados representativos (podem ser, por exemplo: 100000, 200000, 500000, 750000, 1000000). Para os gerar pode: (1) criar um array de tamanho N que vai de 0 a N; (2) remover um qualquer elemento (com exceção do primeiro e último); (3) baralhar o array.
- ⇒ Só deve ser contabilizado o tempo do algoritmo (exclui-se o tempo de leitura/geração do input e de impressão dos resultados).
- ⇒ Sugere-se a realização de várias medições (pelo menos 5) para cada solução e apresentação da média.
- ⇒ Devem apresentar e discutir as regressões para as 3 soluções, incluindo também o coeficiente de determinação/regressão (R^2).

Tabela para as 3 soluções

| <i>n</i> | <i>ALG. 1 (ms)</i> | <i>ALG. 2 (ms)</i> | <i>ALG. 3 (ms)</i> | <i>Iterações</i> |
|----------------|--------------------|--------------------|--------------------|------------------|
| <i>100</i> | 0,004 | 0,006 | 0,001 | 100 |
| <i>500</i> | 0,067 | 0,045 | 0,002 | 100 |
| <i>1000</i> | 0,273 | 0,102 | 0,003 | 10 |
| <i>10000</i> | 14,005 | 0,903 | 0,012 | 10 |
| <i>100000</i> | 1678,024 | 10,617 | 0,105 | 10 |
| <i>250000</i> | 12562,532 | 28,726 | 0,256 | 10 |
| <i>500000</i> | 43371,003 | 62,425 | 0,573 | 10 |
| <i>750000</i> | 113941,976 | 96,078 | 0,854 | 10 |
| <i>1000000</i> | 270186,144 | 130,887 | 1,081 | 10 |
| <i>2000000</i> | 546003,387 | 326,691 | 2,164 | 10 |

Gráfico para a solução A

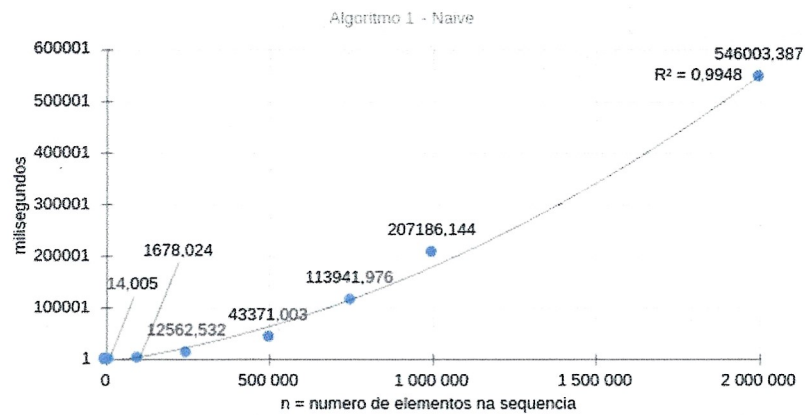


Gráfico para a solução B

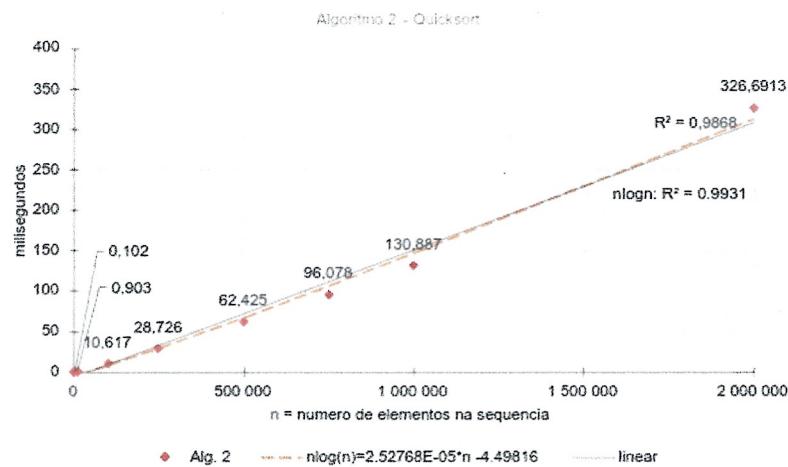
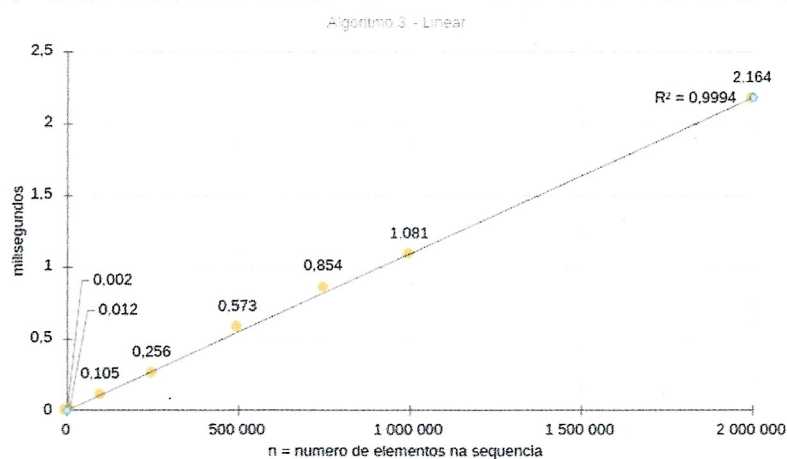


Gráfico para a solução C



Análise dos resultados tendo em conta as regressões obtidas e como estas se comparam com as complexidades teóricas:

O primeiro e o último algoritmo seguiram as regressões com coeficientes superiores a 99%. O primeiro algoritmo, chamado de "naive" ou "bruteforce" tem uma complexidade Temporal de $O(n^2)$, pois no pior caso tem que iterar ~~em~~ o array inteiro para cada elemento. O R^2 de 0.9948 para a regressão $an^2 + b$ confirma esta hipótese.

O terceiro algoritmo implementa a soma de ~~todos~~ ^{e compara} o mínimo e máximo dos valores no array, ~~comparando~~ com a soma dos elementos do array, o que requer uma só iteração do array independente da ordem dos elementos. O valor 0.9999 do coeficiente de regressão confirma a complexidade linear $O(n)$.

O segundo algoritmo implementa quicksort via `qsort` da `library glibc` para encontrar o elemento que falta. $O(n \log n)$ é a complexidade teórica pois a procura ($\frac{1}{2}n$) não é relevante. O gráfico apresenta duas recursões pois, para este conjunto de dados $T \leq 2,000,000$ aparenta ser linear. O coeficiente de regressão para a recursão $n \log n$ é, embora de forma ligeira, menor que o coeficiente da recursão linear. O algoritmo 2 ~~se~~ tem a sua complexidade teórica suportada pelos dados obtidos.