Relatório Projeto 2 AED 2021/2022

Nome: João Miguel Fernandes Moura
Nº Estudante:2020235800
PL (inscrição): 7 *Login* no *Mooshak:2020235800*

Tabela para as 3 soluções

	Solução A(segundos)	Solução B(segundos)	Solução C(segundos)
25000	69.54749608039856	0.00300002098083496	0.00300097465515136
50000	288.5750198364258	0.00600099563598632	0.00500082969665527
75000	647.1604714393616	0.0100352764129638	0.00700211524963378
100000	1136.4331188201904	0.0140125751495361	0.00900125503540039
125000	4480.2062265872955	0.0180006027221679	0.0120110511779785

Gráfico para a solução A

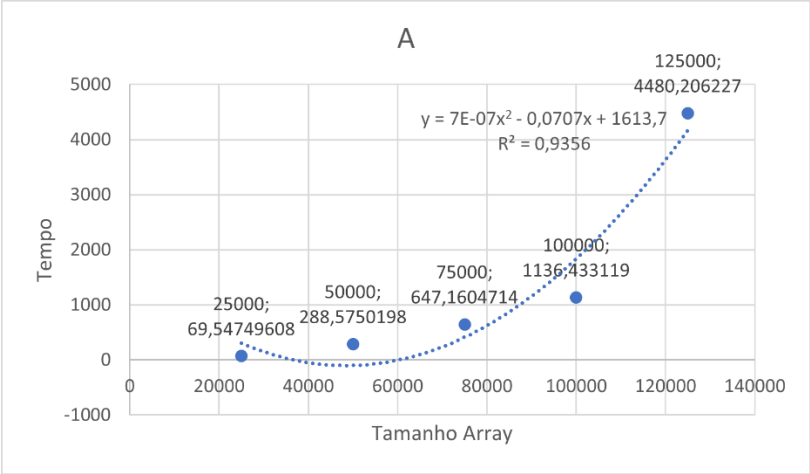


Gráfico para a solução B

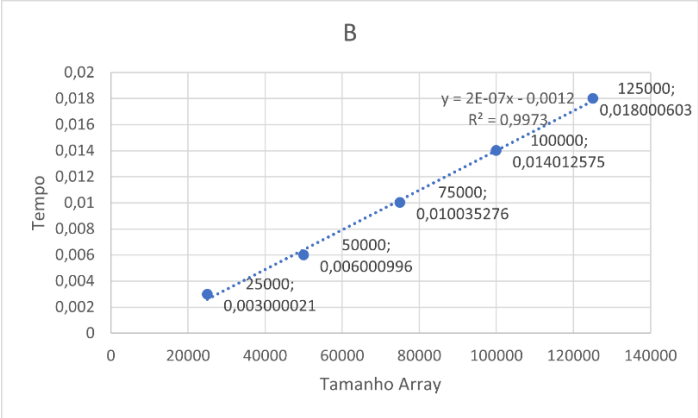
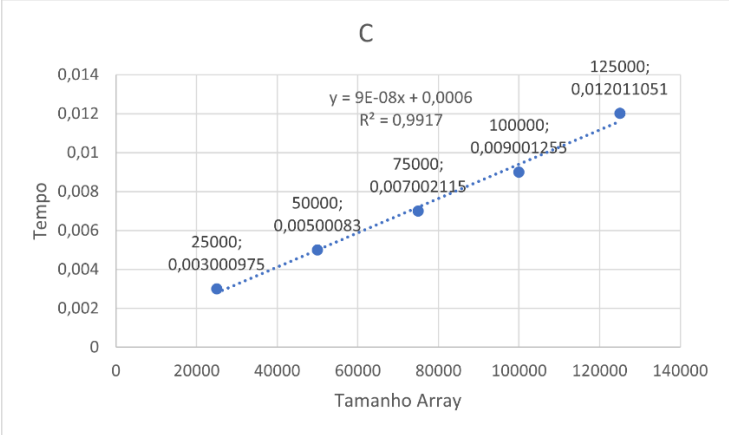


Gráfico para a solução C



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

O gráfico A representa uma evidente regressão polinomial sendo o polinómio de grau 2 logo a complexidade é quadrática (N^2) como estava previsto teoricamente. A complexidade desta solução exaustiva provem da utilização de dois ciclos *for*, um dentro do outro sendo que cada ciclo tem complexidade N (linear). Esta é a solução mais lenta e menos eficiente, tornando-se impraticável nos casos em que se usam milhares de dados de entrada, como verificam os dados com base nos *datasets* utilizados (para 125000 elementos no *array* demora 4480 segundos).

O gráfico B evidencia uma clara regressão linear dos resultados obtidos, no entanto, a complexidade prevista teoricamente é Nlog(N). Como o *dataset* fornecido é limitado não é possível ver o momento em que a tendência se afasta do linear, logo podemos verificar que para *datasets* reduzidos esta solução tem uma taxa de crescimento próxima de linear. A previsão teórica do Nlog(n) é justificada pela utilização da função integrada do *Python .sort()* que tem essa complexidade. O algoritmo *sort* utiliza um ciclo forçosamente, dado que percorre o *array* todo uma vez, que tem complexidade N. No entanto o algoritmo *TimSort* utilizado pela função *sort* do *Python* não faz iterações utilizando dois ciclos um dentro do outro, ao invés, analisa os dados contidos no *array* através de *Runs* (blocos de dados do array), logo a complexidade não será quadrática (N^2) mas antes log(n). A solução B é mais eficiente que a solução A e para um dataset reduzido é quase tão eficiente como a solução C.

O gráfico C evidencia uma regressão linear dos resultados obtidos, o que corresponde ao previsto teoricamente, dado que esta solução só utiliza apenas um ciclo *for* sendo este de complexidade N. Esta complexidade verifica-se independentemente da quantidade de dados de entrada. Esta solução é a mais

eficiente das três soluções estudadas e que apresenta resultados notoriamente mais rápidos isto devido a ser a única das soluções que utiliza apenas um ciclo *for*.

Após a análise das taxas de crescimento obtidas, concluímos o que está previsto teoricamente, que N<Nlog(n)<N^2, sendo a solução C a única viável de utilizar para grandes quantidades de dados.