

# Relatório Projeto 1 AED 2023-2024

Nome: Eduardo Luís Pereira Marques  
PL (inscrição): PL1

Nº Estudante: 2022231584

Registrar os tempos computacionais das 3 soluções. Os tamanhos das arrays (N) devem ser: 20000, 40000, 60000, 80000, 100000. Só deve ser contabilizado o tempo do algoritmo. Exclui-se o tempo de leitura do input e de impressão dos resultados. Devem apresentar e discutir as regressões para as 3 soluções, incluindo também o coeficiente de determinação/regressão ( $r$  quadrado).

Tabela para as 3 soluções

Tamanho do array (N)	Média do tempo decorrido (segundos)		
	Solução exaustiva	Solução c/ ordenamento	Solução elaborada
20000	8,37128	0,00189	0,0003
40000	34,85824	0,00379	0,0007
60000	76,67143	0,00624	0,00093
80000	143,66028	0,00692	0,00133
100000	224	0,01021	0,00168

Gráfico para a solução A

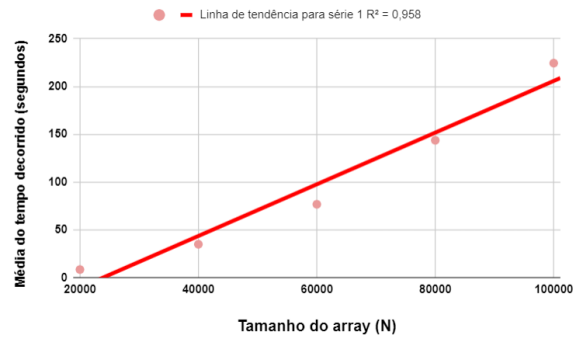


Gráfico para a solução B

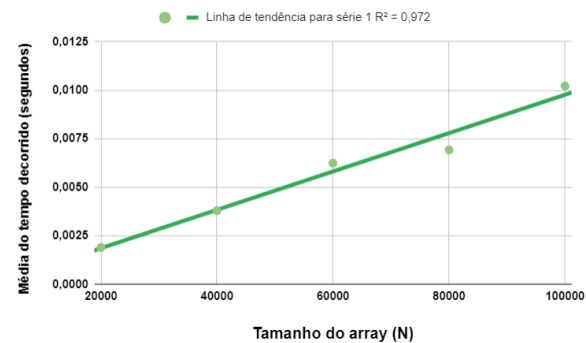
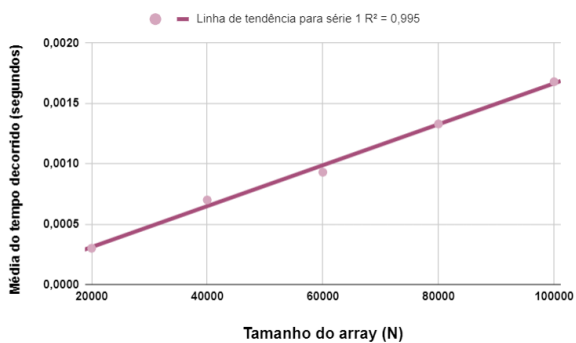


Gráfico para a solução C



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

A solução exaustiva apresentou um aumento significativo no tempo de execução, evidenciado pela sua regressão linear e apontou o valor de  $r$  quadrado mais baixo. Em contrapartida, a solução com ordenamento apresentou um nível de eficiência maior e os tempos decorridos baixaram drasticamente. Por fim, a solução elaborada foi a que melhor respondeu ao problema, demorando menos tempo e exibindo um notável valor para o  $r$  quadrado, o que permite a previsão de futuras tendências, destacando-se assim pela positiva. Ainda assim, os três algoritmos apresentam valores para o  $r$  quadrado bastante satisfatórios o que indica que as regressões lineares se ajustam bem aos dados, ou seja, existe uma boa relação entre o tamanho do array e o tempo de execução, mas diferem na precisão e na rapidez com que alcançam o resultado pretendido. Conclui-se que a solução elaborada é a mais eficaz, seguida pela solução com ordenamento enquanto que a solução exaustiva apresentou o pior desempenho.

## Código:

```
import random
import time

def algoritmo1(k, lista):
    outputExaustiva = 'False'
    inicioExaustiva = time.time()
    for i in lista:
        for j in lista:
            if (i + j == k and i != j):
                output = 'True'
    fimExaustiva = time.time()
    tempoExaustiva = fimExaustiva - inicioExaustiva
    return outputExaustiva, tempoExaustiva

def algoritmo2(k, lista):
    outputOrdenada = 'False'
    ini = 0
    fim = -1
    inicioOrdenada = time.time()
    lista.sort()
    for i in range(len(lista)):
        soma = lista[ini] + lista[fim]
        if (soma == k and lista[ini] != lista[fim]):
            output = 'True'
            break
        elif (soma < k):
            ini += 1
        elif (soma > k):
            fim -= 1
    fimOrdenada = time.time()
    tempoOrdenada = fimOrdenada - inicioOrdenada
    return outputOrdenada, tempoOrdenada
```

```
def algoritmo3(k, lista):
    outputElaborada = 'False'
    temporaria = set()
    inicioElaborada = time.time()
    for i in lista:
        if i < k:
            if k - i in temporaria and i != k - i:
                output = 'True'
                break
            else:
                temporaria.add(i)
    fimElaborada = time.time()
    tempoElaborada = fimElaborada - inicioElaborada
    return outputElaborada, tempoElaborada

def media(total, repeticoes):
    media = format(total/repeticoes, ".5f")
    return media
```

```
def main():
    tamanhos = [20000, 40000, 60000, 80000, 100000]
    repeticoes = 10
    dicio = {}
    dicio['Exaustiva'] = {}
    dicio['Ordenamento'] = {}
    dicio['Elaborada'] = {}
    for i in tamanhos:
        totalExaustiva = 0
        totalOrdenada = 0
        totalElaborada = 0
        for j in range(repeticoes):
            lista = []
            k = random.randint(1, 100)
            for v in range(i):
                lista.append(random.randint(1, 100))

            outputExaustiva, tempoExaustiva = algoritmo1(k, lista)
            totalExaustiva += tempoExaustiva

            outputOrdenada, tempoOrdenada = algoritmo2(k, lista)
            totalOrdenada += tempoOrdenada

            outputElaborada, tempoElaborada = algoritmo3(k, lista)
            totalElaborada += tempoElaborada

        dicio['Exaustiva'][str(i)] = media(totalExaustiva, repeticoes)
        dicio['Ordenamento'][str(i)] = media(totalOrdenada, repeticoes)
        dicio['Elaborada'][str(i)] = media(totalElaborada, repeticoes)
    print(dicio)

if __name__ == '__main__':
    main()
```