

Relatório Projeto 1 AED 2023-2024

Nome: Sérgio Lopes Marques

Nº Estudante: 2022222096

PL (inscrição): 1

Registar 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^2 quadrado).

Tabela para as 3 soluções

Tamanho do Array	Tempo Médio (s)		
	Exaustiva	Ordenada	Elaborada
20000	11,42594731	0,002217722	0,000994039
40000	46,27509558	0,003768039	0,001813507
60000	104,1565731	0,009894919	0,002591777
80000	215,1589043	0,010904336	0,003246069
100000	370,8212035	0,012612343	0,004242063

Gráfico para a solução A

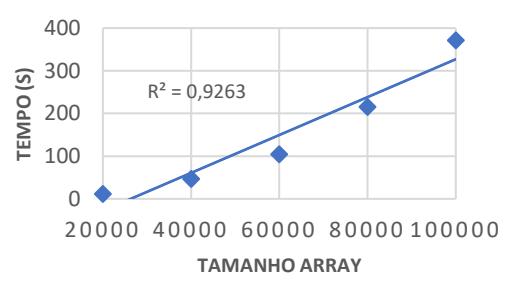


Gráfico para a solução B

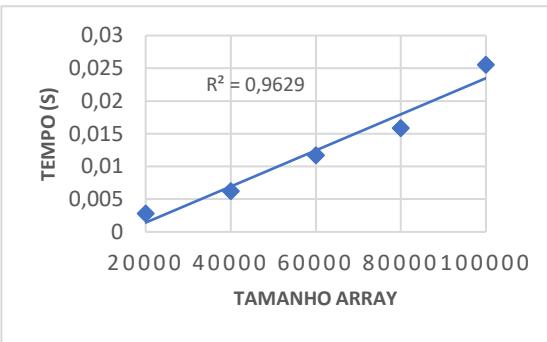
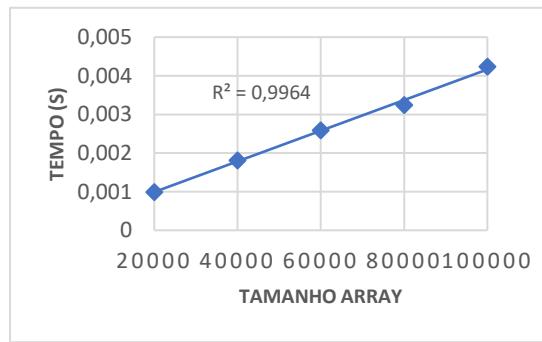


Gráfico para a solução C



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

Analisando os resultados obtidos para os 3 algoritmos em estudo, são perceptíveis as diferenças entre eles. A que mais sobressai é a diferença entre a solução exaustiva (A) e as restantes soluções. Esta solução é a menos eficiente, uma vez que apresenta tempos de execução incomportáveis em comparação com os tempos calculados para as 2 outras soluções. Ao olharmos para o gráfico que contém a regressão linear desta solução, encontramos indícios de alguma ineficiência, que são confirmados pelo seu coeficiente de regressão, uma vez que é o mais baixo dos 3 algoritmos. Em relação às outras duas soluções, podemos afirmar que a diferença entre elas é relativamente pequena. A eficiência destes algoritmos não é colocada em questão, ao serem duas boas soluções para o problema apresentado. Contudo, a solução elaborada (C) é, sem dúvida, a mais eficiente, apresentando tempos de execução muito próximos de 0 segundos e, ainda, uma regressão linear com um coeficiente de regressão muito próximo de 1, o que revela que esta solução é a que mais se adequa ao problema apresentado, independentemente do tamanho do array de elementos utilizado.

Código

Solução A

```
def exaustiva(lista, k):

    output = 'False'
    tempoInicio = time.time()

    for i in lista:
        for j in lista:
            if i + j == k and i != j:
                output = 'True'

    tempoFim = time.time()
    tempo = tempoFim - tempoInicio

    return output, tempo
```

Solução B

```
def ordenada(lista, k):

    output = 'False'
    tempoInicio = time.time()

    lista.sort()
    v = 0
    n = len(lista) - 1

    for i in range(len(lista)):
        if lista[v] + lista[n] == k and lista[v] != lista[n]:
            output = 'True'
            break
        elif lista[v] + lista[n] > k:
            n -= 1
        elif lista[v] + lista[n] < k:
            v += 1

    tempoFim = time.time()
    tempo = tempoFim - tempoInicio

    return output, tempo
```

Solução C

```
def elaborada(lista, k):

    output = 'False'
    tempoInicio = time.time()

    temp = set()

    for valor in lista:
        if valor < k:
            comp = k - valor
            if comp in temp and valor != comp:
                output = 'True'
                break
            temp.add(valor)

    tempoFim = time.time()
    tempo = tempoFim - tempoInicio

    return output, tempo
```

Main

```
def main():

    tamLista = 20000
    n = 5
    numTestes = 10

    for z in range(n):

        mediaExaustiva = 0
        mediaOrdenada = 0
        mediaElaborada = 0

        print("==", tamLista, "==\n")

        for i in range(numTestes):

            k = random.randint(1, 100)
            lista = []
            for j in range(tamLista):
                lista.append(random.randint(1, 100))

            outputExaustiva, tempoExaustiva = exaustiva(lista, k)
            mediaExaustiva += tempoExaustiva
            outputOrdenada, tempoOrdenada = ordenada(lista, k)
            mediaOrdenada += tempoOrdenada
            outputElaborada, tempoElaborada = elaborada(lista, k)
            mediaElaborada += tempoElaborada

        mediaExaustiva /= numTestes
        mediaOrdenada /= numTestes
        mediaElaborada /= numTestes

        print("Média Exaustiva (", tamLista, "): ", mediaExaustiva, "\n")
        print("Média Ordenada (", tamLista, "): ", mediaOrdenada, "\n")
        print("Média Elaborada (", tamLista, "): ", mediaElaborada, "\n\n")

    tamLista += 20000

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