Relatório Projeto 2 AED 2023-2024

Nome: Daniel Coelho Pereira Nº Estudante: 2021237092
PL (inscrição): 4

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 quadrado).

Tabela para as 3 soluções

Alg\tamanho do array	20000	40000	60000	80000	100000
Alg 1	11.610556	46.838962	105.135138	193.025522	301.727522
Alg 2	0.0022044	0.0041908	0.006195	0.008406	0.009799
Alg 3	0.0001900	0.0004000	0.000800	0.001199	0.001395

(tempos em segundo)

Gráfico para a solução A

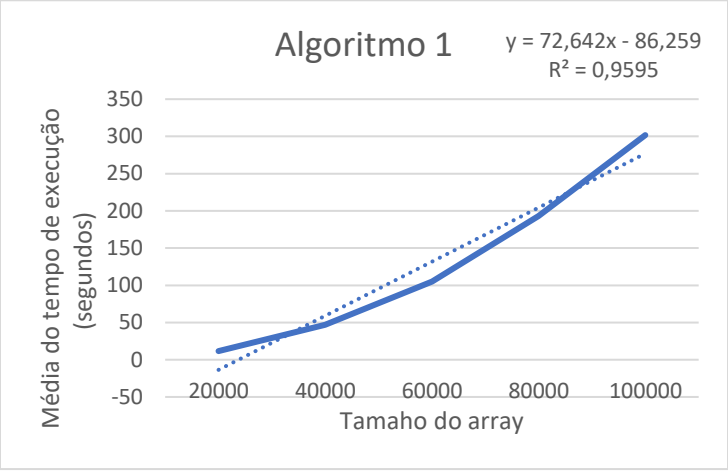


Gráfico para a solução B

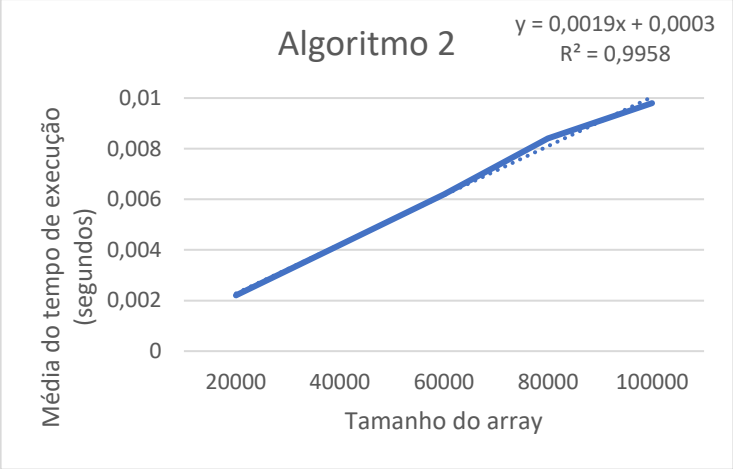
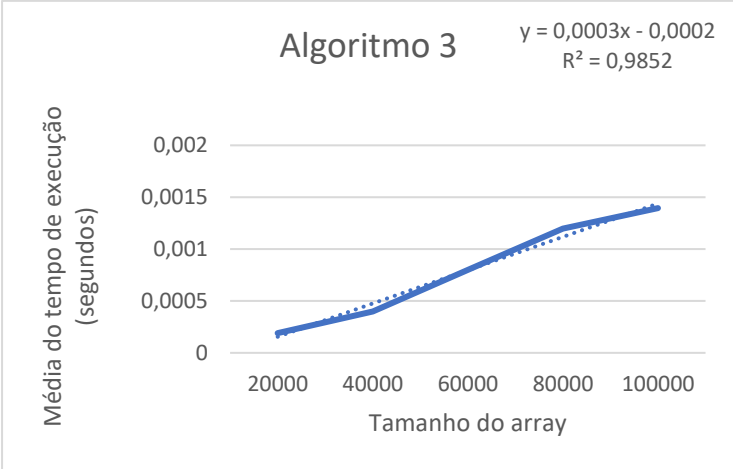


Gráfico para a solução C



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

Através da análise dos resultados notamos facilmente uma redução significativa no tempo de execução nos algoritmos B e C em comparação com o A. Sendo o algoritmo C, como esperado, o mais eficiente.

Interpretando as regressões, verificamos que a o coeficiente angular vai diminuindo (72.642, 0.0019, 0.0003) ou seja a inclinação reduz.

Temos o seguinte resultado se compararmos as regressões com as complexidades teóricas :

A - $O(n^2)$, B - $O(n \log(n))$, C - $O(n)$

Relatório Projeto 2 AED 2023-2024

Código em Python adaptado para facilitar a recolha dos tempos de execução:

```python
import time
import random
def gerar_array(tamanho):
    return [random.randint(0, 99) for _ in range(tamanho)]

def algoritmo1(array,k):
    v = False
    tempo_inical = time.time()
    for i in range(len(array)):
        for x in range(len(array)):
            if array[i] + array[x] == k and array[i] != array[x]:
                v = True
    tempo_final = time.time()
    duracao = tempo_final - tempo_inical
    return v, duracao

def algoritmo2(array,k):
    v = False
    x = 0
    j = len(array) - 1
    tempo_inical = time.time()
    array.sort()
    while array[x]!=array[j]:
        if array[x]+array[j] > k:
            j-=1
        elif array[x] + array[j] < k:
            x+=1
        elif array[x] + array[j] == k and array[x] != array[j]:
            v = True
            break
    tempo_final = time.time()
    duracao = tempo_final - tempo_inical
    return v,duracao

def algoritmo3(array,k):
    v = False
    complementar = set()
    tempo_inical = time.time()
    for i in range(len(array)):
        x = k - array[i]
        if x in complementar and x != array[i]:
            v = True
            break
        else:
            complementar.add(array[i])
    tempo_final = time.time()
    duracao=tempo_final - tempo_inical
    return v, duracao


def main():
    tamanhos = [20000,40000,60000,80000,100000]
    repeticoes=5
    for i in tamanhos:
        soma_algoritmo1=0
        soma_algoritmo2=0
        soma_algoritmo3=0
        for j in range(repeticoes):
            array=gerar_array(i)
            k=random.randint(1,100)
            v_algoritmo1,duracao_algoritmo1=algoritmo1(array,k)
            soma_algoritmo1+=duracao_algoritmo1
            v_algoritmo2,duracao_algoritmo2=algoritmo2(array,k)
            soma_algoritmo2+=duracao_algoritmo2
            v_algoritmo3,duracao_algoritmo3=algoritmo3(array,k)
            soma_algoritmo3+=duracao_algoritmo3
        media_algoritmo1=soma_algoritmo1/repeticoes
        media_algoritmo2=soma_algoritmo2/repeticoes
        media_algoritmo3=soma_algoritmo3/repeticoes
        print("ALGORITMO_1 : ", i, media_algoritmo1)
        print("ALGORITMO_2 : ", i, media_algoritmo2)
        print("ALGORITMO_3 : ", i, media_algoritmo3)

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