

Relatório Projeto 2 AED 2023-2024

Nome: Tomás Pinto Loureiro
PL (inscrição): 8

Nº Estudante: 2022234440

Tabela para as 3 soluções

	20000	40000	60000	80000	100000
Solução A	181.5	567.5	1249.8	2321.1	3642.9
Solução B	3.1	13.8	17.4	22.2	26.3
Solução C	0.19	0.84	1.12	1.53	1.71

Tabela em milissegundos

Gráfico para a solução A

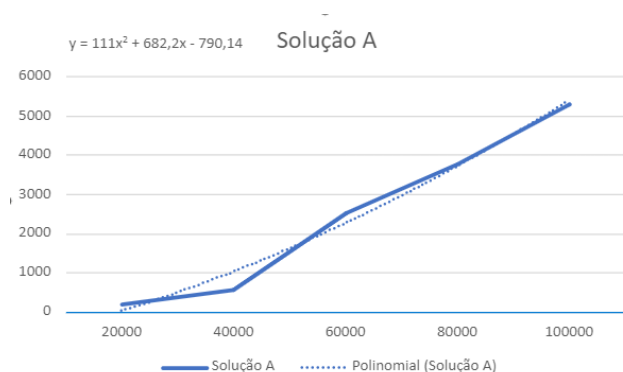


Gráfico para a solução B

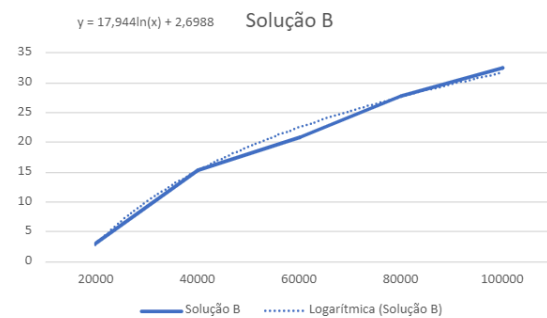


Gráfico com regressão

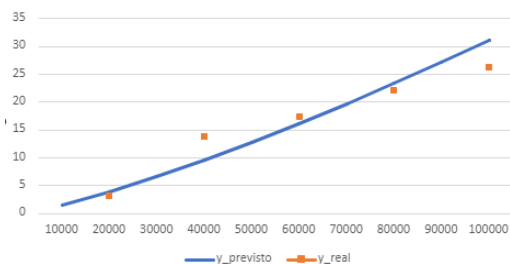
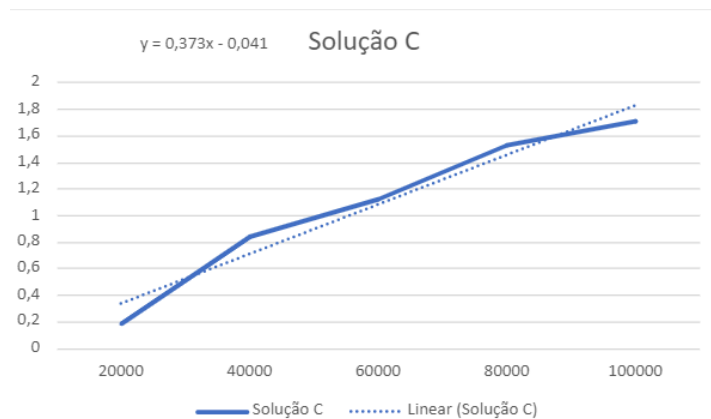


Gráfico para a solução C



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

Na solução 1 que é a solução exaustiva podemos observar que utilizando um regressão polinomial encontramos a função $111x^2 + 682,2x - 790,14$ e o $R^2 = 0,953636144$ e com um grau de complexidade de n^2 .

Na solução 2 utilizando uma regressão logarítmica encontramos a função $17,944 \ln(x) + 2,6944$ e calculando uma previsão para os resultados podemos encontrar uma variação de mais ou menos 10% de acordo com os resultados que obtive. O $R^2 = 0,949812128$ e com um grau de complexidade de $n \log n$.

Na solução 3 utilizando uma regressão linear encontramos a função $0,0373x - 0,041$ e o $R^2 = 0,959060578$. O grau de complexidade é de n .

CÓDIGO:

Solução 1:

```
public static boolean exercicio1(List<Integer> list, int k){
    boolean resposta = false;
    for(int i = 0; i<list.size();i++){
        for(int j = i+1; j<list.size(); j++){
            if (list.get(i) + list.get(j) == k && list.get(i) != list.get(j)){
                resposta = true;
            }
        }
    }
    return resposta;
}
```

Solução 2:

```
public static boolean exercicio2(List<Integer> list, int k){
    int i = 0;
    int j = list.size()-1;
    Collections.sort(list);
    while (i < j){
        if ((list.get(i) + list.get(j) == k) && (list.get(i) != list.get(j))){
            return true;
        }
        else if(list.get(i) + list.get(j) < k && (list.get(i) != list.get(j))){
            i++;
        }
        else if(list.get(i) + list.get(j) > k && (list.get(i) != list.get(j))){
            j--;
        }
    }
    return false;
}
```

Solução 3:

```
public static boolean exercicio3(List<Integer> list, int k){
    List<Integer> list0 = new ArrayList<>(Collections.nCopies(Collections.max(list), 0:0));
    for(int i = 0; i<list.size(); i++){
        if(list.get(i)<k ){
            if(list0.get(k-list.get(i)) != 0 && k != (list.get(i)*2)){
                return true;
            }
            else{
                list0.set(list.get(i), element:1);
            }
        }
    }
    return false;
}
```

CÓDIGO MAIN:

```
public static void main(int size){
    Random rand = new Random();
    List<Integer> list = new ArrayList<Integer>();
    int k;
    long startTime;
    long endTime;
    long time = 0;
    long startTime2;
    long endTime2;
    long time2 = 0;
    long startTime3;
    long endTime3;
    long time3 = 0;

    for(int i = 0; i<10000; i++){
        list = new ArrayList<Integer>();
        list = criarArray(list, size, rand);
        k = rand.nextInt(2*size) + 1;
        startTime = System.nanoTime();
        exercicio1(list, k);
        endTime = System.nanoTime();
        time = time + endTime - startTime;
        startTime2 = System.nanoTime();
        exercicio2(list, k);
        endTime2 = System.nanoTime();
        time2 = time2 + endTime2 - startTime2;
        startTime3 = System.nanoTime();
        exercicio3(list, k);
        endTime3 = System.nanoTime();
        time3 = time3 + endTime3 - startTime3;
        System.out.println(i);
    }
    time = time/100;
    time2 = time2/10000;
    time3 = time3/10000;
    System.out.println("Size " + size + ", Exercicio 1: " + time + ", Exercicio2: " + time2 + ", Exercicio 3: " + time3);
}
```

Neste código a solução 1 é apenas repetido 100 vezes separadamente (com o uso de comment lines para não executar a 2 e a 3) já a solução 2 e 3 é repetida 10000 vezes por questões de tempo.

Código executado:

```
public static void main(String[] args) {
    main(size:20000);
    main(size:40000);
    main(size:60000);
    main(size:80000);
    main(size:100000);
}
```