

lista 02

Parte I

Questão 1 (Laço simples). Considere o seguinte algoritmo:

```
ls ()  
1      aux = 1  
2      i ← 0  
3      enquanto i < n faça  
4          aux = aux + i  
5          i ← i + 1  
6      retorne aux
```

Conte quantas vezes cada uma das linhas é executada.

Questão 2 (Trocando listas de lugar). Imagine que temos duas listas, L e V trocar os elementos de uma lista para outra, isto é, os valores da lista L vão para a lista V e vice-versa:

L:

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 16 | 19 | 41 | 8 | 42 | 39 | 33 | 22 | 29 | 48 |
|----|----|----|---|----|----|----|----|----|----|

V:

| | | | | | | | | | |
|---|---|----|---|----|----|----|----|----|----|
| 2 | 5 | 17 | 9 | 35 | 19 | 23 | 72 | 91 | 44 |
|---|---|----|---|----|----|----|----|----|----|

⇓

L:

| | | | | | | | | | |
|---|---|----|---|----|----|----|----|----|----|
| 2 | 5 | 17 | 9 | 35 | 19 | 23 | 72 | 91 | 44 |
|---|---|----|---|----|----|----|----|----|----|

V:

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 16 | 19 | 41 | 8 | 42 | 39 | 33 | 22 | 29 | 48 |
|----|----|----|---|----|----|----|----|----|----|

```
troca_listas()  
  i ← 0  
  enquanto i < n faça  
    aux = L[i]  
    L[i] = V[i]  
    V[i] = aux  
    i ← i + 1
```

Calcule o número de execuções de instruções (linhas) desse algoritmo.

Parte II

Questão 3 (Múltiplos de 3). Considere uma lista de inteiros L de tamanho n

L:

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 16 | 19 | 41 | 8 | 42 | 39 | 33 | 22 | 29 | 48 |
|----|----|----|---|----|----|----|----|----|----|

Considere o seguinte algoritmo que percorre a lista e sempre que encontra um número múltiplo de 3, divide o número por 3 e eleva ao quadrado.

```
    multiplo_tres()
1      i ← 0
2      enquanto i < n faça
3          se 3 | L[i] então
4              L[i] = L[i] * 3
5              L[i] = L[i] * L[i]
6          i ← i + 1
```

(Considere que o tamanho da lista n seja um múltiplo de 3.)

- a) Quantas vezes a linha 6 será executada?
- b) Se a quantidade de elementos divisíveis por 3 for d , quantas vezes a linha 4 será executada?
- c) Se a quantidade de elementos divisíveis por 3 for d , qual o total de execuções de instruções do algoritmo?
- d) No melhor e no pior caso, quais os números de execuções de instruções realizadas?

Questão 4 (Triplidade). Considere uma lista de inteiros L de tamanho n

L:

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 16 | 19 | 41 | 8 | 42 | 39 | 33 | 22 | 29 | 48 |
|----|----|----|---|----|----|----|----|----|----|

Considere o seguinte algoritmo que percorre a lista e realiza operações sobre os elementos da lista de acordo com o resto da divisão do elemento por 3.

```

triplicidade()
1      i ← 0
2      enquanto i < n faça
3          se 3 % L[i] = 0 então
4              L[i] = L[i] * 3
5              L[i] = L[i] * L[i]
6          senão
7              se 3 % L[i] = 1 então
8                  imprimir('Uhuuuuu!')
9              senão
10                 se 3 % L[i] = 2 então
11                     L[i] = i * L[i]
12                     L[i] = L[i]/3
13                     L[i] = L[i] - 1
14      i ← i + 1

```

Suponha que a quantidade de elementos divisíveis por 3 é x_0 , os que deixam resto 1 na divisão por 3 é x_1 , e a quantidade de elementos que deixam resto 2 na divisão por 3 é x_2 (consequentemente, $x_0 + x_1 + x_2 = n$).

- Quantas vezes a linha 2 será executada?
- Quantas vezes a linha 7 será executada?
- Qual o total de execuções de instruções do algoritmo?
- No melhor e no pior caso, quais os números de execuções de instruções realizadas?

Parte III

Questão 5 (Elementos mais distantes). Considere uma lista de inteiros L de tamanho n .

L :

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 16 | 19 | 41 | 8 | 42 | 39 | 33 | 22 | 29 | 48 |
|----|----|----|---|----|----|----|----|----|----|

Encontre os elementos que estão mais distantes um do outro, ou seja, o valor absoluto da diferença é máximo.

Exemplo: Na lista acima o elementos mais afastados são 8 e 48.

Questão 6 (Segundo maior). Considere uma lista de inteiros L de tamanho n .

L :

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 16 | 19 | 41 | 8 | 42 | 39 | 33 | 22 | 29 | 48 |
|----|----|----|---|----|----|----|----|----|----|

Encontre o segundo maior elemento do vetor.

Exemplo: no caso da lista acima, o segundo maior elemento é 42.

Você consegue fazer isso percorrendo o vetor apenas uma vez?

Questão 7 (Topos). Dada uma lista de inteiros L , um topo é um elemento $L[i]$ tal que $L[i-1] < L[i] > L[i+1]$.

Escreva um algoritmo que imprime os topos de uma lista dada.

Questão 8 (Topos crescentes). Dada uma lista de inteiros L , um topo é um elemento $L[i]$ tal que $L[i-1] < L[i] > L[i+1]$.

Escreva um algoritmo que imprime a maior sequência crescente de topos começando com o primeiro topo da lista.

Parte IV

Questão 9 (Elemento repetido). Considere uma lista de inteiros L de tamanho n que na qual todos os elementos são diferentes, exceto um que aparece repetido.

L :

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 16 | 19 | 22 | 8 | 42 | 39 | 33 | 22 | 29 | 48 |
|----|----|----|---|----|----|----|----|----|----|

Exemplo: no caso acima, o elemento repetido é o 22.

Escreva um algoritmo que encontra um elemento repetido em uma lista de inteiros.

Questão 10 (Soma). Escreva um algoritmo que, dada uma lista de inteiros L e um inteiro k , verifica se existem dois elementos de L cuja soma é igual a k .

(Dica: teste todo os pares de elementos da lista.)

Questão 11 (Trocas). Considere dois vetores de inteiros L e V .

L :

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 16 | 19 | 41 | 8 | 42 | 39 | 33 | 22 | 29 | 48 |
|----|----|----|---|----|----|----|----|----|----|

V :

| | | | | | | | | | |
|---|---|----|---|----|----|----|----|----|----|
| 2 | 5 | 17 | 9 | 35 | 19 | 23 | 72 | 91 | 44 |
|---|---|----|---|----|----|----|----|----|----|

Considere a seguinte operação de troca: troque o menor elemento de L pelo maior elemento de V .

Escreva um algoritmo que realiza essa operação de troca enquanto o menor elemento de L for menor que o maior elemento de V .

Qual o número máximo de trocas que podem ocorrer? E quantos passos mais ou menos o algoritmo executa no pior caso em função de n (algo parecido com $an + b$ ou $an^2 + bn + c$, ou $n\sqrt{n}, \dots$).