

Noções de complexidade de algoritmos III

(exercícios)

Atividade

Calcule as quantidades de instruções, bem como a complexidade de pior caso (O) dos trechos de programas que se seguem.

Questão 1

`int x = 0;` — — |

`x = x + 1;` — — |

`printf("%d", resultado);` — — |

Resposta: pior caso $O(1)$

Questão 2

```
int numero; — 1
scanf("%d", &numero); — 1
if(numero % 2 == 0) — 3, 1, 1
    numero = numero + 1; + 1
else — 1
    numero = numero - 1; — 0
printf("%d", numero); — 1
```

Resposta: pior caso $O(1)$

Questão 3

```
int max = 0;  _ 1  
int i = 0;    _ 1  
while(i < max) - 0 0  
    i++;      - 0
```

Resposta: $T(n) = 1 + 1 \rightarrow O(1)$

Questão 4

```
int max = 10; - 1  
int i = 0; - 1  
while(i < max) - n + 1  
    i++; - 1
```

Resposta: $1 + 1 + (n+1) + n = O(n)$

Questão 5

```
int max = 10; — 1
```

```
int i = 0; — 1
```

```
while(i < max) — n+1
```

```
    printf("%d", i++); n
```

Resposta: $1 + 1 + (n + 1) + n + n = 3n + 3 \rightarrow O(n)$

Questão 6

```
int max = 0; — |  
for(int i = 0; i < max; i++) — 0 |  
    printf("%d", i); — 0
```

Resposta: $1 + 1 = 2 \rightarrow O(1)$

Questão 7

```
int max = 10; - 1  
for(int i = 0; i < max; i++) - 1 + (n - 1)  
    printf("%d", i); - 1
```

Resposta: $1 + (n + 1) + n + n = 3n + 2 \rightarrow O(n)$

Questão 8

```
int max = 100; |  
int a = 0;  
for(int i = 0; i < max; i++){ -  $1 + (n + 1) + n$   
    a = a + i; -  $n$   
    printf("%d", a); -  $n$   
}
```

Resposta: $1 + (n + 1) + n + n + n = 4n + 2 \rightarrow O(n)$

Questão 9

```
int n;
scanf("%d", &n);
for(int i = 0; i < n; i++){
    printf("%d", i);
}
```

Handwritten notes: $1 + (n + 1) + n$ and a squiggly line under the `printf` statement.

Resposta: $1 + (n + 1) + n + n = 3n + 2 \rightarrow O(n)$

Questão 10

```
int a = 0; — |
```

```
for(int i = 1; i < n; i++){ 1 + (n + 1) + (n - 1)
```

```
    a++;
```

```
    printf("%d", i); ~ — |
```

```
}
```

```
printf("%d", a); — | Resposta:  $1 + (n + 1) + (n - 1) + (n - 1) + 1 = 3n \rightarrow O(n)$ 
```

Questão 11

```
for(int i = 0; i < n; i++)  
    for(int j = 0; j < n; j++)  
        printf("%d", j);
```

Handwritten notes in blue ink:

- Next to the first `for` loop: $1 + (n+1) + (n-1)$
- Next to the second `for` loop: $n * (//)$
- Next to the `printf` statement: $n \wedge 2$

Resposta: $(n + 1) + n * ((n + 1) + n) = n^2 + 2n \rightarrow O(n^2)$

Questão 12

```
int n = 10; 1
```

```
int num[n];
```

```
for(int i = 0; i < n; i++) - 1 + (n + 1) + n
```

```
if(num[i] % 2 == 0) - n
```

```
printf("%d", num[i]); - n
```

Resposta: $1 + (n + 1) + n + n = 3n + 2 \rightarrow O(n)$

Questão 13

```
int n = 5;      - 1
int num[n];     - 1
for(int i = 0; i < n; i++) - 1 + (n + 1) + n
    if(num[i] % 2 == 0) - n
        printf("%d", num[i]); - n
```

Resposta: $1 + 1 + (n + 1) + n + n = 3n + 3 \rightarrow O(n)$

Questão 14

```
int a = n; // n > 3 - 7
```

```
int b = 3; - 1
```

```
do{ - m
```

```
    printf("%d", a*(a - b)); - m =
```

```
    a--; - m - 3
```

```
}while(a > b); - m - 3 + 1
```

$3m - 6 \rightarrow O(m)$

Resposta.

Fim da aula