

Marcos Antônio Lommez Cândido Ribeiro

• Pergunta 1: Qual é a diferença entre as notações O , Ω e Θ ? Pesquise!!!

As notações apresentadas são respectivamente chamadas de

O = Big O

Ω = Big Ômega

Θ = Big Theta

Elas servem para mostrar os diferentes graus de complexidade de crescimento de um algoritmo. Mas todo algoritmo tem sua dificuldade diferenciada entre pior caso, melhor caso e caso médio, que é exatamente o que elas representam, sendo o Big O para o pior, Big Ômega para o melhor e o Big Theta para representar o caso médio.

O mais comum de se representar é o Big O, sendo esta uma representação do tempo máximo de execução do software, mesmo não servindo na prática para representar o tempo real do algoritmo, mas nos apresenta uma abstração sobre o comportamento. O melhor caso sendo este o Ω (Big Ômega) normalmente só é utilizado para fins acadêmicos ou quando este representa uma finalidade, enfim existem momentos certos de utilizá-lo no mundo não acadêmico. Por fim também existe o caso médio representado pelo Θ (Big Theta) que é a complexidade média do algoritmo como o próprio nome diz. Normalmente a média real é o que se espera saber de um algoritmo, o qual é representado pelo Big Theta e calculado com técnicas de estatística para se chegar a um valor próximo da realidade.

• Pergunta 2: Qual é a notação O , Ω e Θ para todos os exercícios feitos na Unidade 1b?

Como todos os exercícios apresentados e propostos não possuem condição de parada ou nada que modifique a velocidade de execução do algoritmo, as notações O , Ω e Θ acabam possuindo exatamente o mesmo valor, por isso representarei o resultado utilizando em cada momento uma notação diferente para representar que aqui neste ambiente elas são intercambiáveis.

```
-----  
a--;  
a -= 3;  
a = a - 2;
```

$O(1)$

```
-----  
if (a + 5 < b + 3){  
    i++;  
    ++b;  
    a += 3;  
} else {  
    j++;  
}
```

$\Omega(1)$

```
if (a + 5 < b + 3 || c + 1 < d + 3){  
    i++;  
    ++b;  
    a += 3;  
} else {  
    j++;  
}
```

$\Theta(1)$

```
for (int i = 0; i < 4; i++){  
    a--;  
}
```

$O(1)$

```
for (int i = 0; i < n; i++){  
    a--;  
    b--;  
}
```

$\Omega(n)$

```
int i = 0, b = 10;  
while (i < 3){  
    i++;  
    b--;  
}
```

$\Theta(1)$

```
for (int i = 3; i < n; i++){  
    a--;  
}
```

$O(n)$

```
int i = 10;  
while (i >= 7){  
    i--;  
}
```

$\Omega(1)$

```
for (int i = 5; i >= 2; i--){  
    a--;  
}
```

$\Theta(1)$

```
for (int i = 0; i < 5; i++){  
    if (i % 2 == 0){  
        a--;  
        b--;  
    } else {  
        c--;  
    }  
}
```

$O(1)$

```
int a = 10;  
for (int i = 0; i < 3; i++){  
    for (int j = 0; j < 2; j++){  
        a--;  
    }  
}
```

$\Omega(1)$

```
int a = 10, b = 10, c = 10, d = 10;  
for (int i = 0; i < 3; i++){  
    for (int j = 0; j < 2; j++){  
        a--;  
        b--;  
        c--;  
        d--;  
    }  
}
```

$\Theta(1)$

```
for (int i = 0; i < n; i++){  
    for (int j = 0; j < n; j++){  
        a--;  
    }  
}
```

$O(n^2)$

```

int i = 1, b = 10;
while (i > 0){
    b--;
    i = i >> 1;
}
i = 0;
while (i < 15){
    b--;
    i += 2;
}

```

$\Omega(1)$

```

for (int i = 0; i < n; i++)
    for (int j = 0; j < n - 3; j++)
        a *= 2;

```

$\Theta(n^2)$

```

for (int i = n - 7; i >= 1; i--)
    for (int j = 0; j < n; j++)
        a *= 2;

```

$O(n^2)$

```

for (int i = n; i > 0; i /= 2)
    a *= 2;

```

$\Omega(\lg(n))$

```

for (int i = n+4; i > 0; i >>= 1)
    a *= 2;

```

$\Theta(\lg(n))$

```

for (int i = n - 7; i >= 1; i--)
    for (int j = n - 7; j >= 1; j--)
        a *= 2;

```

$O(n^2)$

```

for (int i = n; i > 0; i /= 2)
    a *= 2;

```

$\Omega(\lg(n))$

```
for (int i = n+1; i > 0; i /= 2)
    a *= 2;
```

$\Theta(\lg(n))$

```
for (int i = n+1; i > 1; i /= 2)
    a *= 2;
```

$O(\lg(n))$

```
for (int i = 1; i < n; i *= 2)
    a *= 2;
```

$\Omega(\lg(n))$

```
for (int i = 1; i <= n; i *= 2)
    a *= 2;
```

$\Theta(\lg(n))$

```
for(int i=0; i<n; i++){
    a--;
    b--;
    c--;
}
for(int i=0; i<n; i++)
    for(int j=n; j>0; j--)
        a--;
```

$O(n^2)$

```
for(int i=0; i<n; i++){
    a--;
    b--;
    c--;
    d--;
    e--;
}
for(int i = n; i>0; i--)
    for(int j = n; j>0; j--)
        for(int k = n; k>0; k--)
            a--;
```

$\Omega (n^3)$

```
for(int i =0; i<n; i++)  
    a--;  
for(int i = n; i>0; i/=2)  
    a--;
```

$\Theta (\lg(n))$

```
a--;  
b--;  
c--;  
d--;  
e--;  
for(int i = 0; i<n; i++)  
    for(int j = 0; j<n; j++)  
        for(int k = n; k>0; k--)  
            a--;
```

$O(n^3)$

```
for(int i = n; i>0; i--){  
    for(int j = n; j>0; j--){  
        a--; b--; c--; e--;  
        for(int k = n; k>0; k--){  
            for(int l = n; l>0; l--){  
                a--; b--; c--; d--; e--;  
            }  
        }  
    }  
}  
for(int i = 0; i<n/2; i++)  
    a--;
```

$\Omega (n^4)$

```
for(int i = n; i>0; i/=2)  
    a--;  
for(int i = n; i>0; i/=2){  
    a--; b--; c--; d--; e--;  
}
```

$\Theta (\lg(n))$

```
int min = array[0];  
for (int i = 1; i < n; i++){  
    if (min > array[i]){  
        min = array[i];  
    }  
}
```

```
}
```

$O(n)$

```
int min = array[0];  
for (int i = 1; i < n; i++){  
    if (min > array[i]){  
        min = array[i];  
    }  
}
```

$\Omega(n)$

```
int min = array[0];  
for (int i = 1; i < n; i++){  
    if (min > array[i]){  
        min = array[i];  
    }  
}
```

$\Theta(n)$