

Estrutura de Dados

Adrian G. S. B. Colombo

Pesquisar sobre o Grande Theta e sobre o Grande Ômega

Grande Theta é uma notação matemática padrão que mostra a eficiência de um algoritmo no pior cenário em relação ao seu tamanho de entrada. Para medir a eficiência de um algoritmo, é necessárias duas coisas: complexidade de tempo e complexidade espacial. A notação Grande Theta captura o limite superior para mostrar quanto tempo ou espaço um algoritmo exigiria no pior cenário à medida que o tamanho da entrada aumenta. Geralmente é escrito como:

$$F(n) = O(\text{tamanhoDeEntrada})$$

A complexidade de tempo é determinada levando-se em consideração dois fatores: o tamanho da entrada e a solução do algoritmo.

Exemplo:

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main() {
5.  int sum = 0;
6.  for (int i=0;i<5;i++){
7.  sum = sum+i;
8.  }
9.  cout << "Sum = " << sum;
10. return 0;
11. }
```

Lista de todas as instruções junto com sua contagem de execução:

Operações	Número de Execuções
int sum = 0	1
for (int i=0;i<5;i++)	6
sum = sum+i	5
cout << "Sum = " << sum	1
return 0	1

Cálculo:

$$1 + 6 + 5 + 1 + 1$$

Generalizando esta notação em termos de tamanho de entrada (n) forma esta expressão:

$$\Rightarrow 1 + (n + 1) + n + 1 + 1 \Rightarrow 1 + (n + 1) + n + 1 + 1$$

Após simplificar a expressão acima, a complexidade de tempo final é:

$$2n + 42n + 4$$

Como estima a notação Grande Theta de um algoritmo?

Para encontrar a notação Grande Theta, é necessário seguir duas etapas: descarte as constantes principais e ignore os termos de ordem inferior. Depois de realizar as duas etapas na complexidade de tempo, pode estimar a notação Grande Theta como:

$$\Rightarrow 2n + 4 \Rightarrow 2n + 4 _ \Rightarrow n + 4 \Rightarrow n + 4 \Rightarrow n \Rightarrow n \Rightarrow O(n) \Rightarrow O(n)$$

Lista de complexidades Grande Theta classificadas em ordem crescente:

Constante	um número constante de operações
Logarítmica	$\log n$
Linear	n
Quadrática	n^2
Cúbica	n^3
Polinomial	n^k , onde k é uma constante
Exponencial	e

Semelhante à notação Grande Theta, a notação Grande Ômega é usada para descrever o desempenho ou a complexidade de um algoritmo. Se um tempo de execução é $\Omega(f(n))$, então para n grande o suficiente, o tempo de execução é pelo menos $k \cdot f(n)$ para alguma constante k. Veja como pensar em um tempo de execução que é $\Omega(f(n))$:

O tempo de execução é Grande Ômega de f(n). Usa a notação Grande Ômega para limites inferiores assintóticos, uma vez que limita o crescimento do tempo de execução a partir de baixo para tamanhos de entrada grandes o suficiente.

A diferença entre a notação Grande Theta e a notação Grande Ômega, é que Grande Theta é usado para descrever o tempo de execução do pior caso para um algoritmo. Mas, a notação Grande Ômega, por outro lado, é usada para descrever o melhor caso de tempo de execução para um determinado algoritmo.