Estrutura de Dados II

Matheus Vicente Delunardo de Lucena
Baseado nas aula de Berilhes Borges Garcia

1 Introdução

- **O que aprenderemos neste curso?** Algoritmos.
- **Que tipo de algoritmos?** Algoritmos de todos os tipos, mas principalmente de ordenação e pesquisa

Algorimos são o espírito da Computação. Computadores podem ser feitos de muitas coisas diferentes. Atualmente eles sao feitos de silício, mas no futuro talve seja feitos de ácido desoxiribonucleico, ou quem sabe, balas de M&M. De qualquer forma o espírito permanecerá inalterado. E é esse espírito que nós chamamos de algoritmo.

Neste curso veremos vários algorítmos, mas nós tamém veremos que existem um conjunto, na verdade um conjunto muito pequeno, de estratégias para construção de algorítmos. E o nosso foco, em grande parte, será em aprender essas técnicas.

- **Dividir e Conquistar**
- **Programação Dinâmica**
- **Abordagem Gulosa**
- **Transforme e Conquiste**
- **Randomização**

2 Bibliografia

2.1 Livro Texto

- "Algoritmos", Cormen, Leisesson, Rivest Stein

2.2 Referências adicionais

- "The Art of Computer Programming", D.E. Knuth, Vol. 3
- "Algorithms", R. Sedgewick (Possui versões em pseudocódigo, C, Pascal e Java)

3 Insertion Sort

3.1 Pseudocódigo

```
InsertionSort(array, n)
for i <— 2 to n
    aux <— array[i]
    j <— i - 1
    while(j > 0 and array[j] > aux) do
        array[j + 1] <— array[j]
        j <— j - 1
    array[j +  1] <— aux
```

3.2 Javascript

```
function insertSort(array) {
    const n = array.length;
    for (let i = 1; i < n; i++) {
        let aux = array[i];
        let j = i - 1;
        while (j >= 0 && array[j] > aux) {
            array[j + 1] = array[j];
            j = j - 1;
        }
        array[j + 1] = aux;
    }
}
const list = [10, 3, 5, 1, 2, 9, 6];
insertSort(list);
console.log("Lista Ordenada", list);
// Lista Ordenada: [1, 2, 3, 5, 6, 9, 10]
```

3.3 Análise

O que devemos utilizar para medir o tempo de execução dos algorítmos? Comparações e atribuições.

Quantas comparações o Insertion Sort faz para n elementos no melhor caso?

$$n - 1 \tag{1}$$

Quantas comparações o Insertion Sort faz para n elementos no melhor caso?

$$\sum_{i=2}^{n} i - 1 \tag{2}$$

Quantas comparações o Insertion Sort faz para n elementos no caso médio?

Essa já é uma pergunta mais complicada, e para repondê-la, precisamos analisar a distribuição dos valores da entrada. Nosso modelo de probabilidade assumirá que todas as entradas são igualmente prováveis, ou seja, tem $1/n$ chances de ocorrer.

Definimos que:

I_n = O número médio de comparações que o Insertion Sort faz para ordenar um vetor aleatório de n elementos.

Com isso, podemos começar a analisar cada caso.

3.3.1 Vetor com 0 elementos

Quanto $n = 0$, ele não fará nenhuma comparação, logo:

$I_0 = 0$

3.3.2 Vetor com 1 elemento

Quando $n = 1$, como só há um elemento, ele não fará nenhuma comparação, logo:

$I_1 = I_0 + 0$

3.3.3 Vetor com 2 elementos

Quando $n = 2$, teremos duas situações:

1. Menor na segunda posição (Probabilidade: $\frac{1}{2}$)
   Vai comparar com a primeira posição, trocar e parar

2. Maior na segunda posição (Probabilidade: $\frac{1}{2}$)
   Vai comparar com a primeira posição e parar

$I_2 = I_1 + \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1$

3.3.4 Vetor com 3 elementos

Quando $n = 2$, teremos três situações:

1. Menor na terceira posição (Probabilidade: $\frac{1}{3}$)
   Vai comparar com a segunda posição, trocar, comparar com a primeira, trocar e parar

2. Segundo menor na terceira posição (Probabilidade: $\frac{1}{3}$)
   Vai comparar com a segunda posição, trocar, comparar com a primeira, e parar

3. Maior na terceira posição (Probabilidade: $\frac{1}{3}$)
   Vai comparar com a segunda posição e parar

$I_3 = I_2 + \frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 1$

Já é possivel identificar um padrão nesses resultados de I_n, então podemos expressá-lo como:

$$I_n = I_{n-1} + \frac{n-1}{n} + \sum_{k=1}^{n-1} \frac{k}{n} \tag{3}$$

$$I_n = I_{n-1} + \frac{n-1}{n} + \frac{1}{n} \sum_{k=1}^{n-1} k \tag{4}$$

$$I_n = I_{n-1} + \frac{n-1}{n} + \frac{1}{n} \cdot \frac{n(n-1)}{2} \tag{5}$$

$$I_n = I_{n-1} + \frac{n-1}{n} + \frac{1}{n} \cdot \frac{n(n-1)}{2} \tag{6}$$

$$I_n = I_{n-1} + \frac{n^2+n-2}{2n} \tag{7}$$

Agora podemos finalmente responder a pergunta deste tópico, quantas comparações o Insertion Sort faz para n elementos no caso médio?

$$I_n = \sum_{k=2}^{n} \frac{k^2+k-2}{2k} \tag{8}$$

Este algoritmo, portanto, tem complexidade média quadratica