Projeto e Análise de Algorítmos I

Ordenando em O(n)

Matheus Gabriel

Novembro 2024

Contents

1 Ordenando em O(n)

Lembrando que o teorema original dizia:

> Qualquer algoritmo de ordenação por comparação requer $\omega(n \lg n)$ comparações no pior caso.

1.1 Problema de ordenação

1.1.1 Enunciado

Escreva um algoritmo de ordenação para ordenar números inteiros de 1 até 99999, não repetidos. Este algoritmo deve rodar em $O(n)$ para o pior caso.

1.1.2 Minha hipótese

Simplesmente imprima os números de 1 a 99999

1.1.3 Uma das soluções

1. Algoritmo para Ordenar Lista de 1 a 99,999 em O(n)

   Este método ordena uma lista contendo números únicos de 1 a 99,999 em tempo O(n) usando uma lista auxiliar indexada diretamente.

   (a) Passos para Implementação

      i. Inicialize uma lista auxiliar de tamanho 99,999. Cada posição representa um número na faixa de 1 a 99,999.

      ii. Percorra a lista original e insira cada número na posição correspondente na lista auxiliar:

         - Exemplo: Coloque o número 1 na posição 0, o número 2 na posição 1, e assim por diante.

      iii. A lista auxiliar estará automaticamente ordenada após a inserção de todos os elementos.

   (b) Implementação pequena em Python

```
import random

def counting_sort(arr):
    # Inicializa o contador com zeros para cada número entre 0 e 9
    count = [0] * 10
    output = [0] * len(arr)

    # Conta a ocorrência de cada elemento na lista
    for num in arr:
        count[num] += 1

    # Atualiza o contador para armazenar as posições dos elementos
    for i in range(1, 10):
        count[i] += count[i - 1]

    # Constrói a lista de saída de forma ordenada
    for num in reversed(arr):
        output[count[num] - 1] = num
        count[num] -= 1

    return output

# Gera uma lista aleatória de 0 a 9 sem repetições
```

```
arr = random.sample(range(10), 10)

# Aplica o Counting Sort
sorted_arr = counting_sort(arr)

return arr, sorted_arr
```

i. Complexidade

- **Tempo**: O(n), pois percorremos a lista uma vez e inserimos cada elemento diretamente na posição.
- **Espaço**: O(n), pois utilizamos uma lista auxiliar de tamanho 99,999.