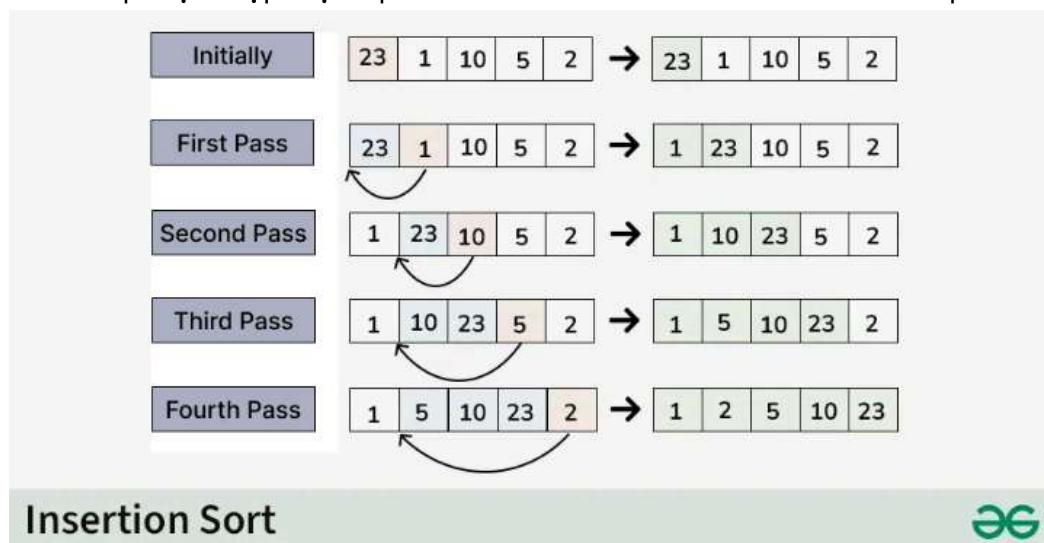


🕒 Cập nhật tháng 8 năm 2024

[Bài Đọc] Giải thuật sắp xếp chèn

1. Khái niệm

- **Insertion Sort** là một thuật toán sắp xếp đơn giản, hoạt động giống như cách bạn sắp xếp các lá bài trong trò chơi bài. Bạn sẽ lần lượt lấy mỗi phần tử trong mảng và chèn nó vào đúng vị trí trong phần đã sắp xếp
- Quá trình hoạt động của Insertion Sort bao gồm:
 - Giả sử rằng phần đầu của mảng đã được sắp xếp
 - Lấy phần tử tiếp theo trong mảng và chèn nó vào vị trí thích hợp trong phần đã sắp xếp
 - Tiếp tục lặp lại quá trình cho đến khi tất cả các phần tử đều được sắp xếp



2. Cách hoạt động

- Giải thuật Insertion Sort sẽ thực hiện qua các bước sau:
 - Bắt đầu từ phần tử thứ hai trong mảng (vì phần tử đầu tiên được xem như là đã sắp xếp)
 - So sánh phần tử hiện tại với các phần tử trước đó trong mảng
 - Dịch chuyển các phần tử lớn hơn phần tử hiện tại một vị trí sang bên phải
 - Chèn phần tử hiện tại vào đúng vị trí đã được tìm thấy
 - Lặp lại bước trên cho đến khi toàn bộ mảng được sắp xếp

3. Đặc điểm

- Độ phức tạp thời gian (Time Complexity):
 - **Tốt nhất:** $O(n)$ (Khi mảng đã được sắp xếp sẵn)
 - **Trung bình và xấu nhất:** $O(n^2)$ (Khi mảng ngược thứ tự hoàn toàn)
- Độ phức tạp không gian (Space Complexity): $O(1)$
- **Tính ổn định:** Insertion Sort là một thuật toán **ổn định**, nghĩa là nếu hai phần tử có giá trị bằng nhau, thứ tự của chúng sẽ không bị thay đổi

4. Ưu và nhược điểm

- Ưu điểm
 - Dễ cài đặt và dễ hiểu
 - Tốt cho danh sách nhỏ hoặc các danh sách gần như đã được sắp xếp
 - Không yêu cầu bộ nhớ bổ sung ($O(1)$)
- Nhược điểm
 - Không hiệu quả đối với các danh sách lớn ($O(n^2)$)
 - Không phải là thuật toán nhanh nhất khi danh sách có nhiều phần tử

5. Cài đặt giải thuật

```

public class InsertionSortExample new *
{
    public static void insertionSort(int[] arr) 1usage new *
    {
        int n = arr.length;

        for (int i = 1; i < n; i++)
        {
            int key = arr[i];
            int j = i - 1;

            // Dịch chuyển các phần tử của arr[0..i-1], mà lớn hơn key
            // Một vị trí lên phía sau của mảng
            while (j >= 0 && arr[j] > key)
            {
                arr[j + 1] = arr[j];
                j = j - 1;
            }

            // Chèn key vào vị trí đúng
            arr[j + 1] = key;
        }
    }

    public static void main(String[] args) new *
    {
        int[] arr = {12, 11, 13, 5, 6};
        System.out.println("Mảng ban đầu:");
        for (int num : arr)
        {
            System.out.print(num + " ");
        }

        insertionSort(arr);

        System.out.println("\nMảng sau khi sắp xếp:");
        for (int num : arr)
        {
            System.out.print(num + " ");
        }
    }
}

```

Link tài nguyên đọc thêm: <https://www.geeksforgeeks.org/insertion-sort-algorithm/>