

🕒 Cập nhật tháng 8 năm 2024

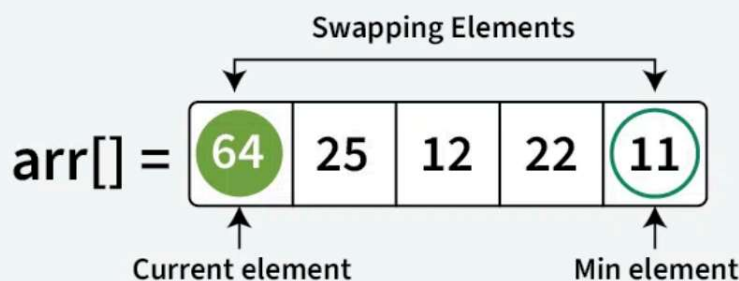
[Bài Đọc] Giải thuật sắp xếp chọn

1. Khái niệm

- Giải thuật sắp xếp chọn (**Selection Sort**) là một thuật toán sắp xếp đơn giản, hoạt động bằng cách chia danh sách thành hai phần:
 - Phần đã được sắp xếp** (ban đầu rỗng)
 - Phần chưa được sắp xếp** (ban đầu chứa toàn bộ danh sách)
- Mỗi lần lặp, thuật toán sẽ tìm phần tử nhỏ nhất (hoặc lớn nhất) từ phần chưa sắp xếp và đưa nó vào cuối phần đã sắp xếp

01
Step

Start from the first element at index 0, find the smallest element in the rest of the array which is unsorted, and swap (11) with current element(64).



Selection Sort Algorithm

2. Cách hoạt động

- Lặp qua danh sách từ đầu đến cuối.
- Trong mỗi lần lặp:
 - Tìm chỉ số của phần tử nhỏ nhất trong phần chưa được sắp xếp
 - Hoán đổi phần tử nhỏ nhất đó với phần tử đầu tiên trong phần chưa được sắp xếp
- Tiếp tục lặp cho đến khi toàn bộ danh sách được sắp xếp

3. Đặc điểm

- Độ phức tạp thời gian (Time Complexity):
 - Tốt nhất, trung bình, và xấu nhất: $O(n^2)$

- Vì thuật toán luôn phải thực hiện $n-1$ phép so sánh trong mỗi vòng lặp bất kể danh sách đã sắp xếp hay chưa
- **Độ phức tạp không gian (Space Complexity):** $O(1)$
- **Tính ổn định:** Sắp xếp chọn **không ổn định**, nghĩa là thứ tự tương đối của các phần tử bằng nhau có thể bị thay đổi

4. Ưu và nhược điểm

- Ưu điểm
 - Dễ cài đặt và hiểu
 - Không yêu cầu bộ nhớ bổ sung
- Nhược điểm
 - Hiệu suất thấp cho danh sách lớn do độ phức tạp $O(n^2)$
 - Ít hiệu quả hơn so với các thuật toán sắp xếp nâng cao như Merge Sort hay Quick Sort

5. Cài đặt giải thuật

```

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

        // Duyệt qua toàn bộ mảng
        for (int i = 0; i < n - 1; i++)
        {
            // Tìm phần tử nhỏ nhất trong phần chưa được sắp xếp
            int minIndex = i;
            for (int j = i + 1; j < n; j++)
            {
                if (arr[j] < arr[minIndex])
                {
                    minIndex = j;
                }
            }

            // Hoán đổi phần tử nhỏ nhất với phần tử đầu tiên
            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;
        }
    }

    public static void main(String[] args) new *
    {
        int[] arr = {64, 25, 12, 22, 11};
        System.out.println("Mảng ban đầu:");
        for (int num : arr)
        {
            System.out.print(num + " ");
        }

        selectionSort(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/selection-sort-algorithm-2/>

