Shaker Sort

Table of contents

Idea

Trong mỗi lần sắp xếp, duyệt mảng theo 2 lượt từ hai phía khác nhau. Lượt đi: đẩy phần tử nhỏ về đầu mảng. Lượt về: đẩy phần tử lớn về cuối mảng. Ghi nhận lại các đoạn đã sắp xếp nhằm tiết kiệm các phép so sánh thừa.

Properties

Shaker Sort là một dạng nâng cao của Bubble Sort nên nó có thể nhận diện được mảng đã sắp xếp. Đồng thời Shaker Sort sẽ tối ưu hơn Bubble Sort trong trường hợp dãy đã gần như có thứ tự.

Ví dụ {2,3,4,5,1} thì Shaker Sort cần 2 lần đi và về, Bubble Sort cần 4 lần duyệt. Tuy nhiên trong trường hợp mảng phân bố ngẫu nhiên thì Shaker Sort có thời gian thực hiện ngang với Bubble Sort

Complexity

Best case và Worst case xảy ra với mảng đầu vào tương tự như Bubble Sort.

Cases	Complexity
Best case	$O(n)$
Worst case	$O(n^2)$
Average case	$O(n^2)$

Space Complexity: $O(1)$.

Code

```
void shakerSort(int *a, int n)
{
    int left = 0, right = n - 1, k = n - 1;
    while(left < right)
    {
        // Đi từ phải qua đẩy nhỏ nhất về đầu mảng
        for(int i = right; i > left; i--)
        {
            if(a[i] <  a[i - 1]){
                swap(a[i], a[i - 1]);
                k = i;//Lưu lại vị trí có hoán vị
            }
        }

        // Giảm kích thước mảng về vị trí có hoán vị
        left = k; // Vì trước đó xem như đã sắp xếp

        // Đi từ trái qua đẩy lớn nhất về cuối mảng
        for(int i = left; i < right; i++)
        {
            if(a[i] > a[i + 1]){
                swap(a[i], a[i + 1]);
                k = i;
            }
        }

        right = k; // Sau đó xem như đã sắp xếp
    }
}
```