SORT

Sắp xếp nổi bọt (Bubble Sort)

Sắp xếp nổi bọt hay bubble sort là thuật toán sắp xếp đầu tiên mà mình giới thiệu đến các bạn và cũng là thuật toán đơn giản nhất trong các thuật toán mà mình sẽ giới thiệu, ý tưởng của thuật toán này như sau:

Duyệt qua danh sách, làm cho các phần tử lớn nhất hoặc nhỏ nhất dịch chuyển về phía cuối danh sách, tiếp tục lại làm phần tử lớn nhất hoặc nhỏ nhất kế đó dịch chuyển về cuối hay chính là làm cho phần tử nhỏ nhất (hoặc lớn nhất) nổi lên, cứ như vậy cho đến hết danh sách Cụ thể các bước thực hiện của giải thuật này như sau:

1. Gán i = 0
2. Gán j = 0
3. Nếu A[j] > A[j + 1] thì đổi chỗ A[j] và A[j + 1]
4. Nếu j < n – i – 1:
   - Đúng thì j = j + 1 và quay lại bước 3
   - Sai thì sang bước 5
5. Nếu i < n – 1:
   - Đúng thì i = i + 1 và quay lại bước 2
   - Sai thì dừng lại

Vd :

https://www.youtube.com/watch?v=fID30q6UWGA

Đánh giá thuật toán sắp xếp bubble sort

Độ phức tạp thuật toán

- Trường hợp tốt: O(n)
- Trung bình: O(n^2)
- Trường hợp xấu: O(n^2)

Không gian bộ nhớ sử dụng: O(1)

Sắp xếp chọn (Selection Sort)

Sắp xếp chọn hay selection sort sẽ là thuật toán thứ hai mà mình giới thiệu đến các bạn, ý tưởng của thuật toán này như sau: duyệt từ đầu đến phần tử kề cuối danh sách, duyệt tìm phần tử nhỏ nhất từ vị trí kế phần tử đang duyệt đến hết, sau đó đổi vị trí của phần tử nhỏ nhất đó với phần tử đang duyệt và cứ tiếp tục như vậy.

Cho mảng A có n phần tử chưa được sắp xếp. Cụ thể các bước của giải thuật này áp dụng trên mảng A như sau:

1. Gán i = 0
2. Gán j = i + 1 và min = A[i]
3. Nếu j < n:
   - Nếu A[j] < A[min] thì min = j
   - j = j + 1
   - Quay lại bước 3
4. Đổi chỗ A[min] và A[i]
5. Nếu i < n – 1:
   - Đúng thì i = i + 1 và quay lại bước 2
   - Sai thì dừng lại

Vd :

Đánh giá thuật toán selection sort

Độ phức tạp thuật toán

- Trường hợp tốt: O(n^2)
- Trung bình: O(n^2)
- Trường hợp xấu: O(n^2)

Không gian bộ nhớ sử dụng: O(1)

Sắp xếp chèn (Insertion Sort)

Sắp xếp chèn hay insertion sort là thuật toán tiếp theo mà mình giới thiệu, ý tưởng của thuật toán này như sau: ta có mảng ban đầu gồm phần tử A[0] xem như đã sắp xếp, ta sẽ duyệt từ phần tử 1 đến n – 1, tìm cách chèn những phần tử đó vào vị trí thích hợp trong mảng ban đầu đã được sắp xếp.

Giả sử cho mảng A có n phần tử chưa được sắp xếp. Các bước thực hiện của thuật toán áp dụng trên mảng A như sau:

1. Gán i = 1
2. Gán x = A[i] và pos = i – 1
3. Nếu pos >= 0 và A[pos] > x:
   - A[pos + 1] = A[pos]
   - pos = pos – 1
   - Quay lại bước 3
4. A[pos + 1] = x
5. Nếu i < n:
   - Đúng thì i = i + 1 và quay lại bước 2
   - Sai thì dừng lại

VD: https://www.youtube.com/watch?v=rHZiWas7z_s

Đánh giá thuật toán sắp xếp chèn

Độ phức tạp thuật toán

- Trường hợp tốt: O(n)
- Trung bình: O(n^2)
- Trường hợp xấu: O(n^2)

Không gian bộ nhớ sử dụng: O(1)

Sắp xếp trộn (Merge Sort)

Sắp xếp trộn (merge sort) là một thuật toán dựa trên kỹ thuật chia để trị, ý tưởng của thuật toán này như sau: chia đôi mảng thành hai mảng con, sắp xếp hai mảng con đó và trộn lại theo đúng thứ tự, mảng con được sắp xếp bằng cách tương tự.

Giả sử left là vị trí đầu và right là cuối mảng đang xét, cụ thể các bước của thuật toán như sau:

- Nếu mảng còn có thể chia đôi được (tức left < right)
  1. Tìm vị trí chính giữa mảng
  2. Sắp xếp mảng thứ nhất (từ vị trí left đến mid)
  3. Sắp xếp mảng thứ 2 (từ vị trí mid + 1 đến right)
  4. Trộn hai mảng đã sắp xếp với nhau

VD: https://www.youtube.com/watch?v=E50u668k_Z8&t=305s

Đánh giá thuật toán sắp xếp merge sort

Độ phức tạp thuật toán

- Trường hợp tốt: O(nlog(n))
- Trung bình: O(nlog(n))
- Trường hợp xấu: O(nlog(n))

Không gian bộ nhớ sử dụng: O(n)

Sắp xếp nhanh (Quick Sort)

Sắp xếp nhanh (quick sort) hay sắp xếp phân đoạn (Partition) là là thuật toán sắp xếp dựa trên kỹ thuật chia để trị, cụ thể ý tưởng là: chọn một điểm làm chốt (gọi là pivot), sắp xếp mọi phần tử bên trái chốt đều nhỏ hơn chốt và mọi phần tử bên phải đều lớn hơn chốt, sau khi xong ta được 2 dãy con bên trái và bên phải, áp dụng tương tự cách sắp xếp này cho 2 dãy con vừa tìm được cho đến khi dãy con chỉ còn 1 phần tử.

Cụ thể áp dụng thuật toán cho mảng như sau:

1. Chọn một phần tử làm chốt
2. Sắp xếp phần tử bên trái nhỏ hơn chốt
3. Sắp xếp phần tử bên phải nhỏ hơn chốt
4. Sắp xếp hai mảng con bên trái và bên phải pivot

Phần tử được chọn làm chốt rất quan trọng, nó quyết định thời gian thực thi của thuật toán. Phần tử được chọn làm chốt tối ưu nhất là phần tử trung vị, phần tử này làm cho số phần tử nhỏ hơn trong dãy bằng hoặc sấp xỉ số phần tử lớn hơn trong dãy. Tuy nhiên, việc tìm phần tử này rất tốn kém, phải có thuật toán tìm riêng, từ đó làm giảm hiệu suất của thuật toán tìm kiếm nhanh, do đó, để đơn giản, người ta thường sử dụng phần tử chính giữa làm chốt.

VD:

https://www.youtube.com/watch?v=qjvx0Ge7aos&fbclid=IwAR1VB1D25ZyGefPaXUxT2hgKP6iC67WgNLSMOXew8Xsep_RiWKZcci-sAhA

Đánh giá thuật toán sắp xếp quick sort

Độ phức tạp thuật toán của quick sort

- Trường hợp tốt: O(nlog(n))
- Trung bình: O(nlog(n))
- Trường hợp xấu: O(n^2)

Không gian bộ nhớ sử dụng: O(log(n))