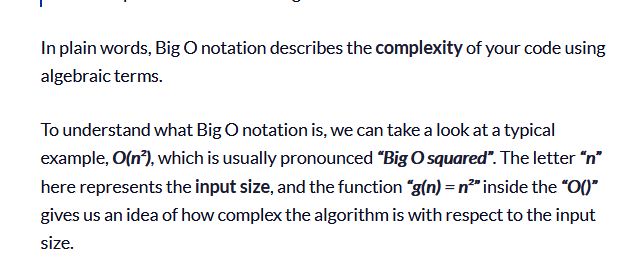
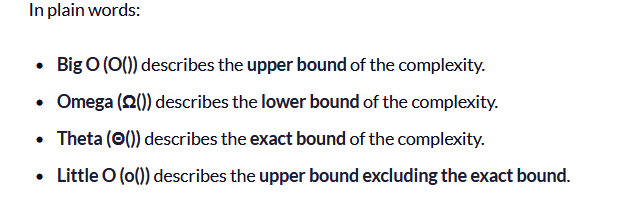
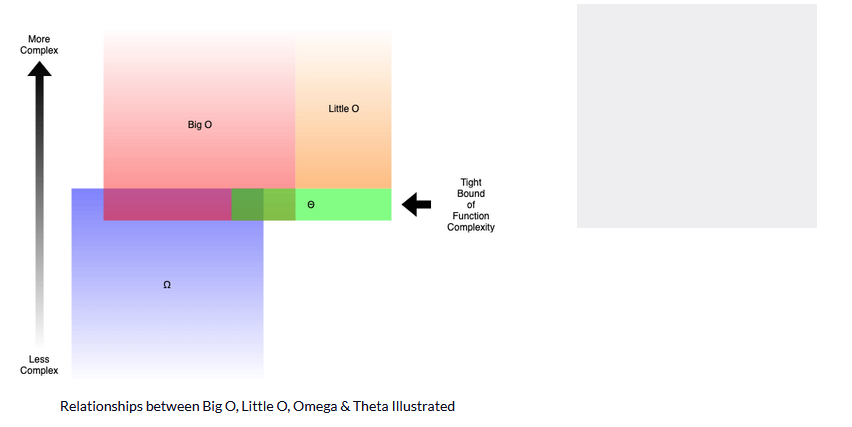


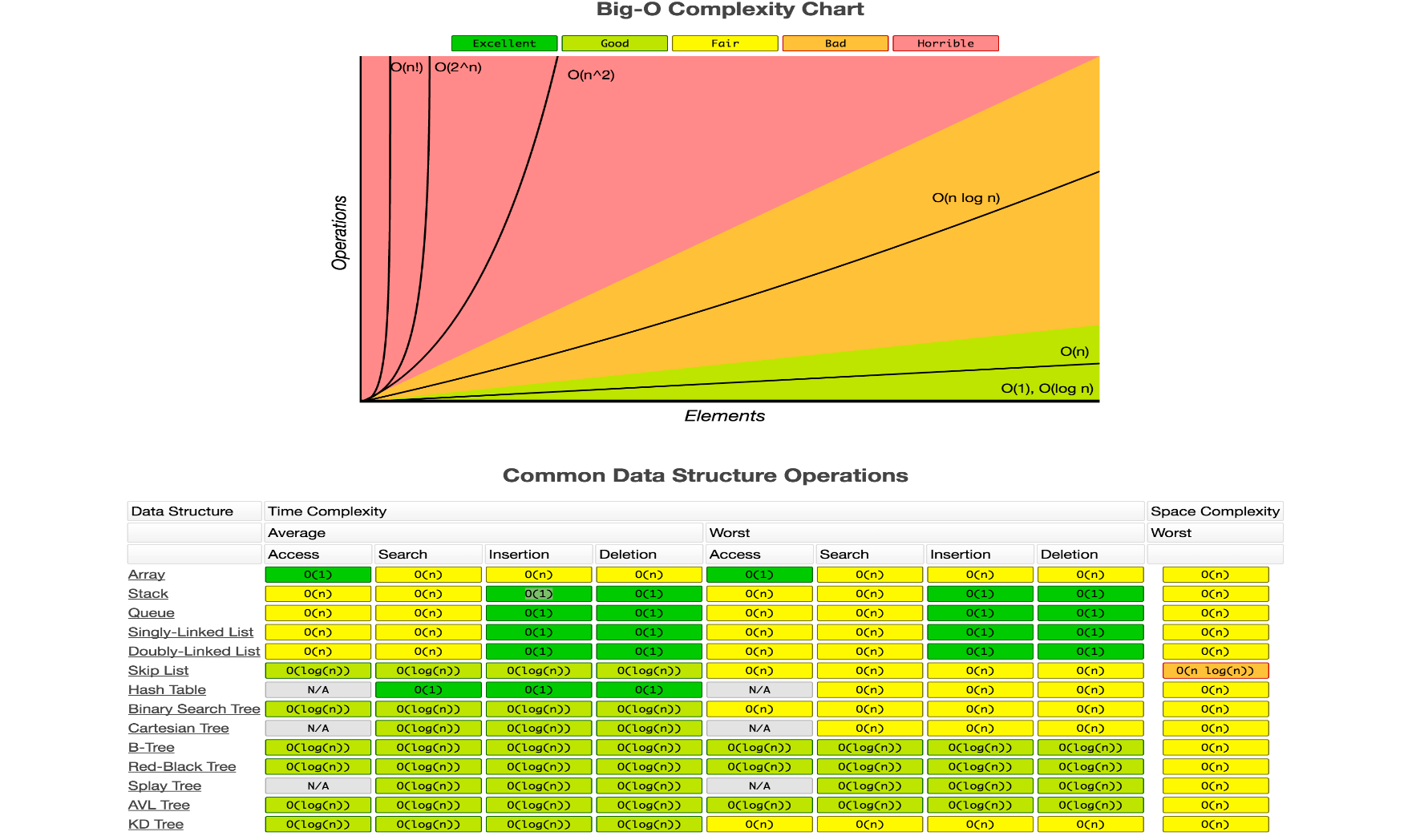
Big O Notation là một ký hiệu toán học mô tả hành vi giới hạn của một hàm khi đối số có xu hướng hướng tới một giá trị cụ thể hoặc vô cùng. Nó là một thành viên của họ ký hiệu được phát minh bởi Paul Bachmann, Edmund Landau và những người khác, được gọi chung là ký hiệu Bachmann – Landau hoặc ký hiệu tiệm cận.

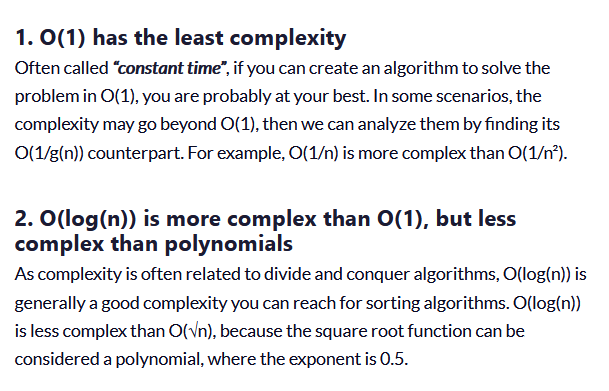
Big O notation is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity. It is a member of a family of notations invented by Paul Bachmann, Edmund Landau, and others, collectively called Bachmann–Landau notation or asymptotic notation.

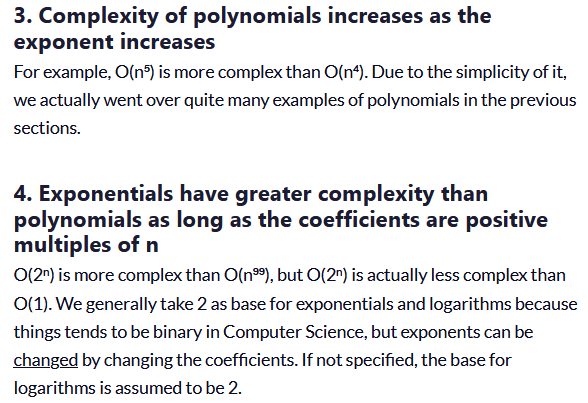


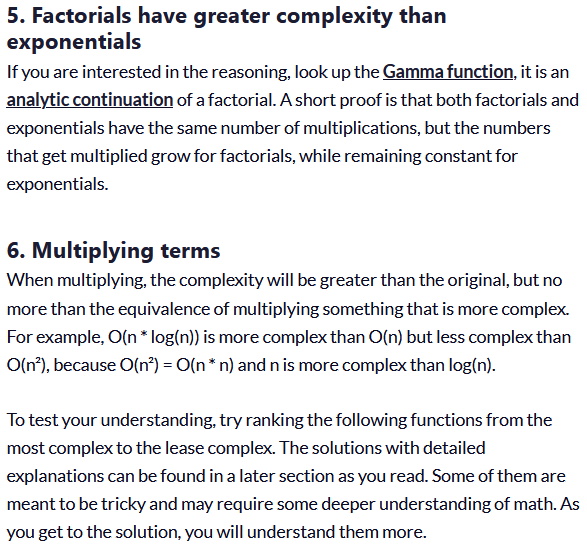




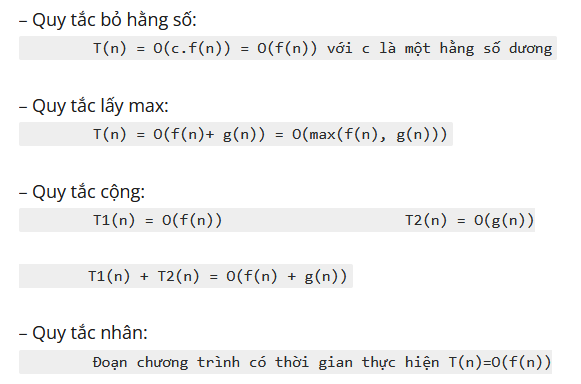




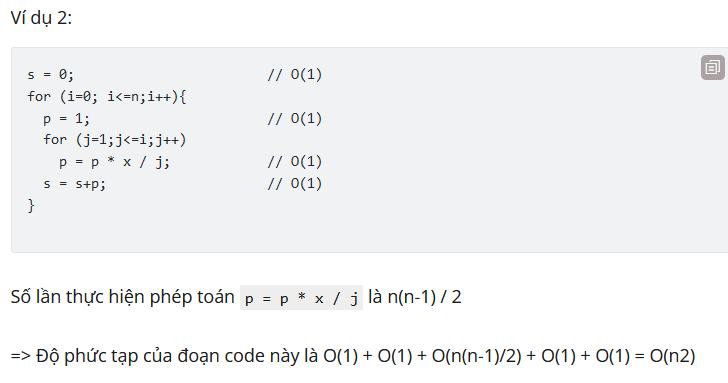


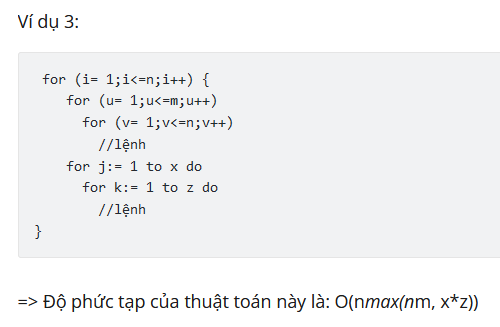


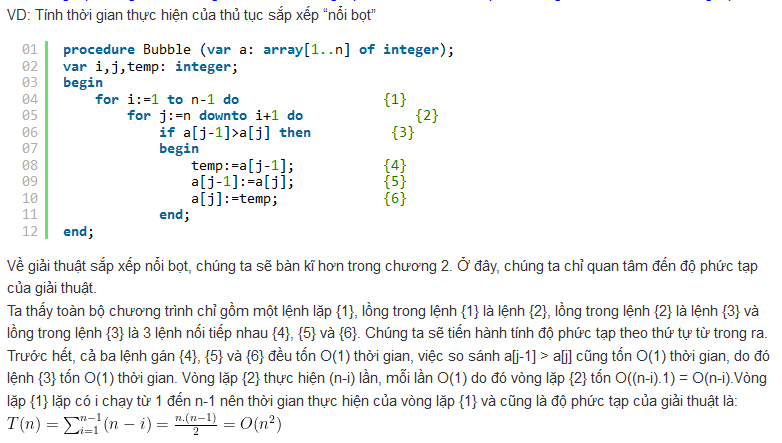


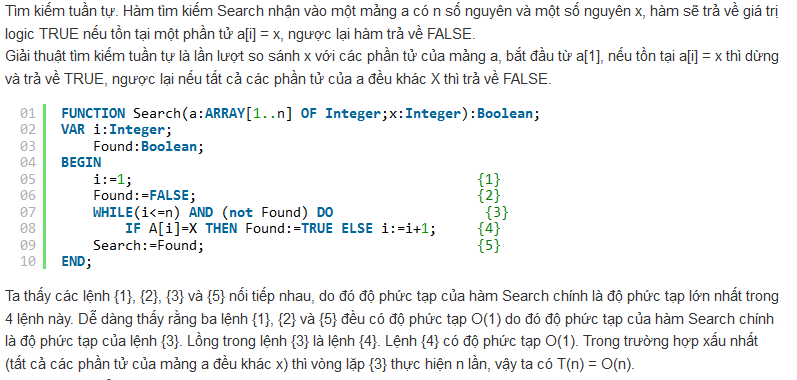


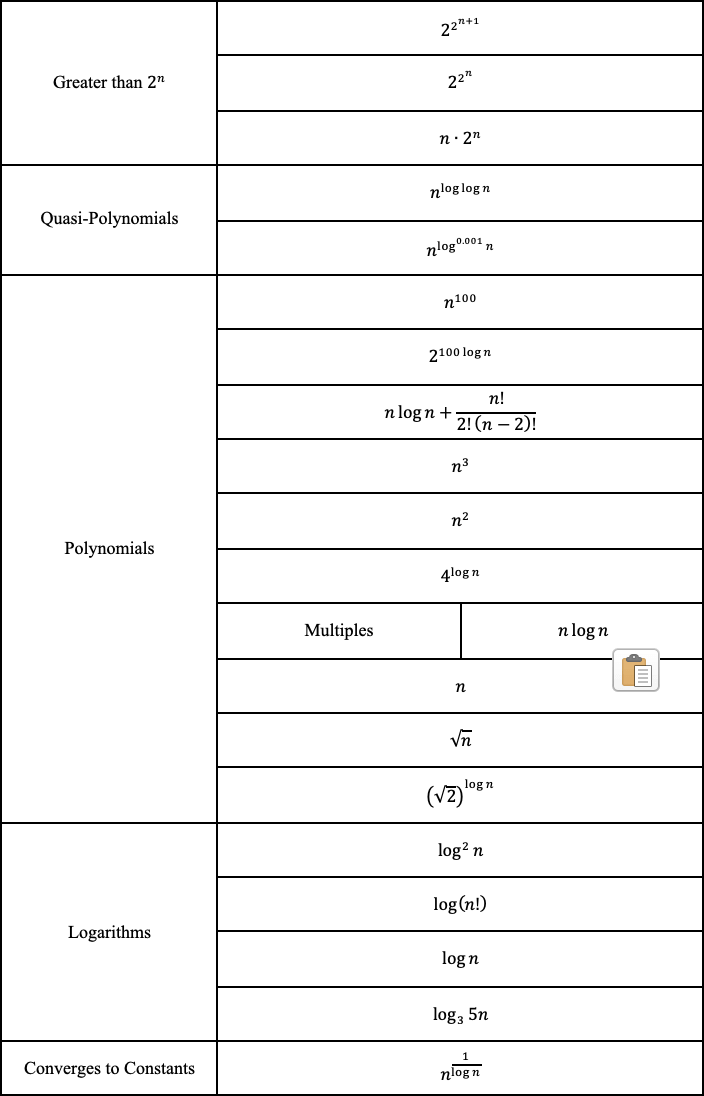
Nếu thực hiện k(n) lần đoạn chương trình với k(n) = O(g(n)) thì độ phức tạp sẽ là O(g(n).k(n))

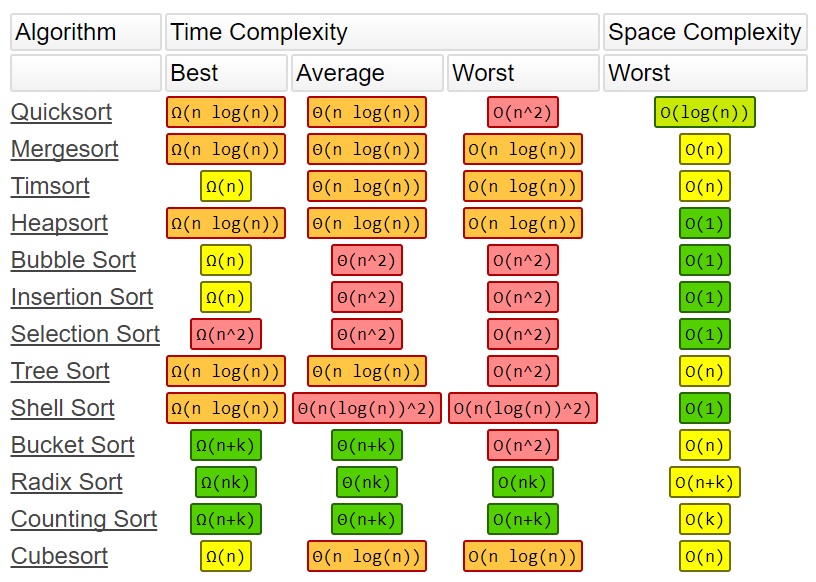












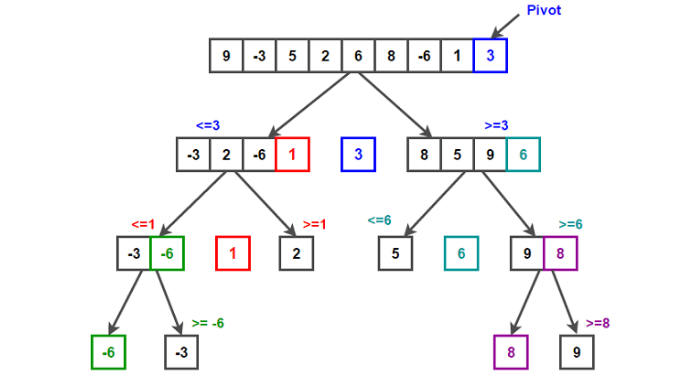
1. **Quick sort (Sắp xếp nhanh)**

Giống như Merge sort, thuật toán sắp xếp quick sort là một thuật toán chia để trị( Divide and Conquer algorithm). Nó chọn một phần tử trong mảng làm điểm đánh dấu(pivot). Thuật toán sẽ thực hiện chia mảng thành các mảng con dựa vào pivot đã chọn. Việc lựa chọn pivot ảnh hưởng rất nhiều tới tốc độ sắp xếp. Nhưng máy tính lại không thể biết khi nào thì nên chọn theo cách nào. Dưới đây là một số cách để chọn pivot thường được sử dụng:

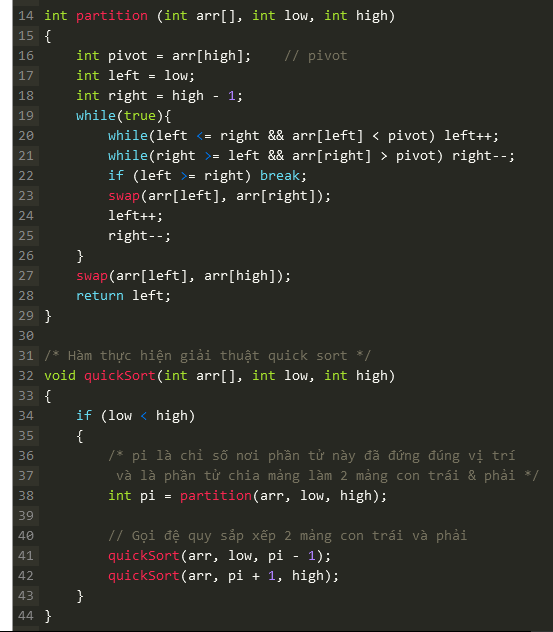
1. Luôn chọn phần tử đầu tiên của mảng.
2. Luôn chọn phần tử cuối cùng của mảng. (Được sử dụng trong bài viết này)
3. Chọn một phần tử random.
4. Chọn một phần tử có giá trị nằm giữa mảng(median element).

Mấu chốt chính của thuật toán quick sort là việc phân đoạn dãy số (Xem hàm partition()). Mục tiêu của công việc này là: Cho một mảng và một phần tử x là pivot. Đặt x vào đúng vị trí của mảng đã sắp xếp. Di chuyển tất cả các phần tử của mảng mà nhỏ hơn x sang bên trái vị trí của x, và di chuyển tất cả các phần tử của mảng mà lớn hơn x sang bên phải vị trí của x.

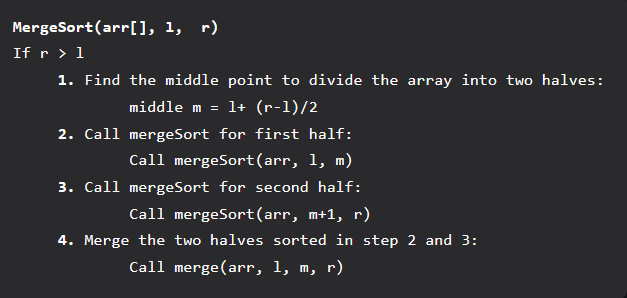
Khi đó ta sẽ có 2 mảng con: mảng bên trai của x và mảng bên phải của x. Tiếp tục công việc với mỗi mảng con(chọn pivot, phân đoạn) cho tới khi mảng được sắp xếp.

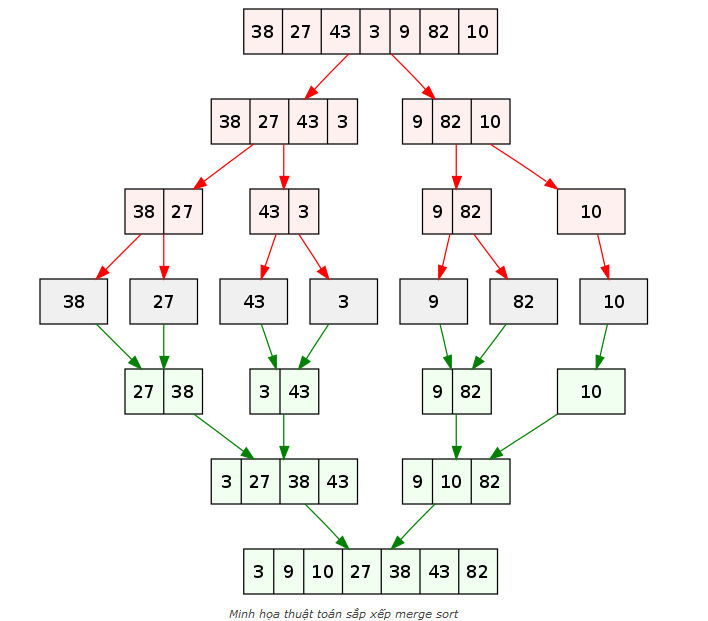


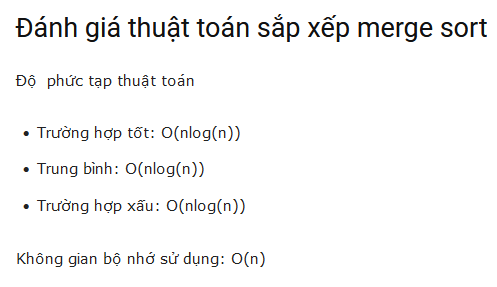




1. **Merge sort (Sắp xếp trộn)**

Like [QuickSort](https://www.geeksforgeeks.org/quick-sort/), Merge Sort is a [Divide and Conquer](https://www.geeksforgeeks.org/divide-and-conquer-introduction/) algorithm. It divides the input array into two halves, calls itself for the two halves, and then merges the two sorted halves. **The merge() function** is used for merging two halves. The merge(arr, l, m, r) is a key process that assumes that arr[l..m] and arr[m+1..r] are sorted and merges the two sorted sub-arrays into one.



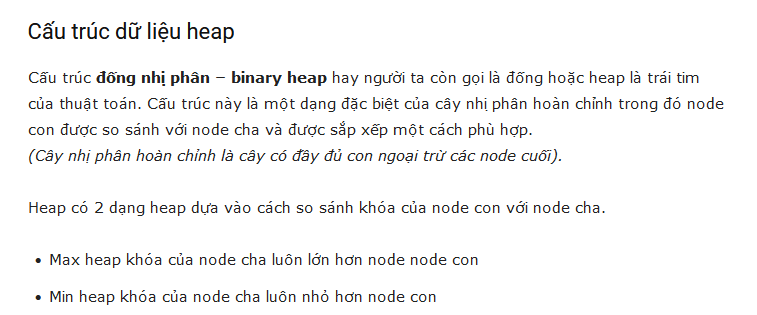


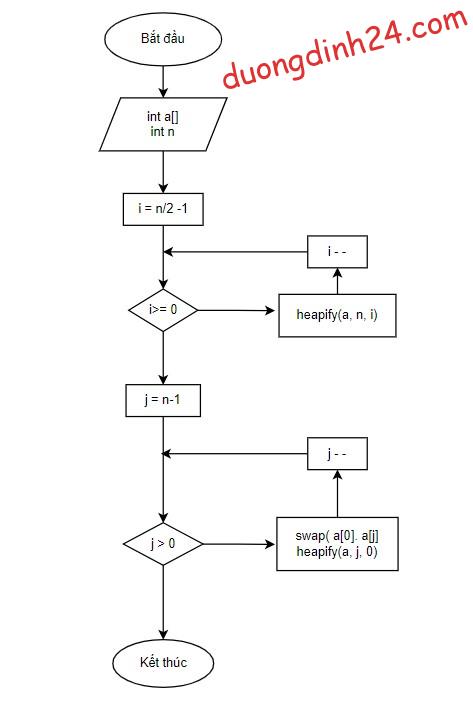
1. **Tim sort**

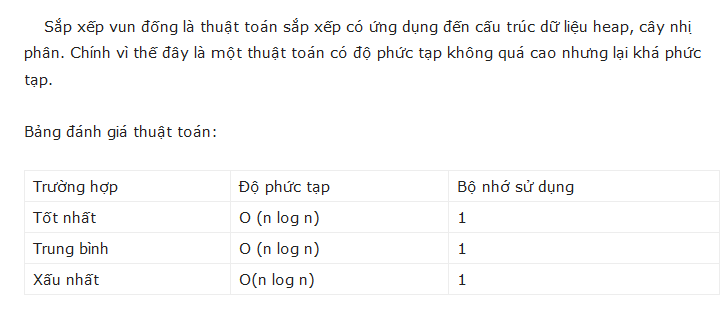
Timsort là một thuật toán sắp xếp hỗn hợp và hiệu quả được tạo ra bằng cách sử dụng khái niệm cả sắp xếp trộn merge sort và sắp xếp chèn inserttion sort.

Hơn nữa, nó được thiết kế để thực hiện tốt trên nhiều loại dữ liệu trong thế giới thực. Tuy nhiên, đây là một trong số ít các thuật toán sắp xếp không xuất hiện trong phòng học. Nó được Tim Peters triển khai vào năm 2002 để sử dụng trong ngôn ngữ lập trình Python. Thuật toán tìm kiếm hệ quả của dữ liệu đã có theo thứ tự và sử dụng kết quả này để sắp xếp phần còn lại hiệu quả hơn.

1. **Heap Sort (Sắp xếp vun đống)**

**Sắp xếp vun đống** – **heap sort** là một thuật toán sắp xếp nhanh sử dụng kĩ thuật phân loại dựa trên cấu trúc cây nhị phân đặc biệt gọi là đống nhị phân (binary heap). Thuật toán dựa vào sự đặc biệt của cây nhị phân để lựa chọn ra phần tử lớn nhất rồi lần lượt chèn phần tử này vào vùng sắp xếp.





1. .
2. .
3. .
4. .
5. .