**Bài 2.5:**

* Chọn:

+ I = 0;

+ j = n - 1 = 7;

+ x = a[ ( I + j ) / 2 ) ] = a[ 3 ] = 60;

* a[ I ] = a[ 0 ] = 40 < x, i++ (I tăng lên 1 giá trị ).
* a[ I ] = a[ 1 ] = 70 > x, dừng ( lúc này I = 1 ).
* a[ j ] = a[ 7 ] = 30 < x, dừng ( lúc này j = 7 ).
* Hoán vị giá trị giữa a[ I ] và a[ j ], tăng I lên 1 giá trị, giảm j xuống 1 giá trị ( I = 2, j = 6 ).
* a[ I ] = a[ 2 ] = 20 < x, i++ ( I tăng lên 1 giá trị ).
* a[ I ] = a[ 3 ] = 60 >= x, dừng ( lúc này I = 3 ).
* a[ j ] = a[ 6 ] = 50 < x, dừng ( lúc này j = 6 ).
* Hoán vị giá trị giữa a[ I ] và a[ j ], tăng I lên 1 giá trị, giảm j xuống 1 giá trị ( I = 4, j = 5 ).
* a[ I ] = a[ 4 ] = 90 > x, dừng ( lúc này I = 4 ).
* a[ j ] = a[ 5 ] = 10 < x, dừng ( lúc này j = 5 ).
* Hoán vị giá trị giữa a[ I ] và a[ j ], tăng I lên 1 giá trị, giảm j xuống 1 giá trị ( I = 5, j = 5 ).
* I = j =5 nên hoán vị giá trị giữa a[ I ] và a[ j ], tăng I lên 1 giá trị, giảm j xuống 1 giá trị ( I = 6, j = 4 ).
* Lúc này I < j, dừng.
* Dãy ban đầu đã được phân thành 2 đoạn con. Mỗi đoạn là 1 dãy con như sau:

+ 40, 30, 20, 50 ( dãy con 1 ).

+ 10, 90, 60, 70 ( dãy con 2 ).

* Trên dãy con 1, ta thực hiện lại việc phân hoạch như trên với vị trí đầu dãy là I = 0, cuối dãy j = 3.
* Trên dãy con 2, ta thực hiện lại việc phân hoạch như trên với vị trí đầu dãy là I = 4, cuối dãy j = 7.
* Tiếp tục thực hiện phân hoạch các dãy con cho đến khi mỗi dãy còn tối đa một phần tử, khi đó danh sách ban đầu được xếp thứ tự:

+ 10, 20, 30, 40, 50, 60, 90.

**Tính độ phức tạp:**

* Ta nhận thấy hiệu quả của thuật toán phụ thuộc vào việc chọn giá trị mốc (hay phần tử chốt).
* Trường hợp tốt nhất: mỗi lần phân hoạch ta đều chọn được phần tử phần tử lớn hơn hay bằng nửa số phần tử và nhỏ hơn hay bằng nửa số phần tử còn lại làm mốc. Khi đó dãy được phân hoạch thành hai phần bằng nhau, và ta cần log2(n) lần phân hoạch thì sắp xếp xong. Ta cũng dễ nhận thấy trong mỗi lần phân hoạch ta cần duyệt qua n phần tử. Vậy độ phức tạp trong trường hợp tốt nhất thuộc O(nlog2(n)).
* Trường hợp xấu nhất: mỗi lần phần hoạch ta chọn phải phần tử có giá trị cực đại hoặc cực tiểu làm mốc. Khi đó dãy bị phân hoạch thành hai phần không đều: một phần chỉ có một phần tử, phần còn lại có n-1 phần tử. Do đó, ta cần tới n lần phân hoạch mới sắp xếp xong. Vậy độ phức tạp trong trường hợp xấu nhất thuộc O(n2).
* Tổng kết lại, ta có độ phức tạp của Quick Sort như sau:
* Trường hợp tốt nhất: O(nlog2n)
* Trường hợp xấu nhất: O(n2)
* Trường hợp trung bình: O(nlog2n)