

## Chapter 3: A One Way, Stackles Quicksort Algorithm

### PHÂN TÍCH CHI TIẾT ĐỒ ÁN CUỐI KỲ

Học kỳ I - Năm học: 2022-2023

—oOo—

Thành viên nhóm

- 21110365 - Thái Bảo An
- 21110445 - Phan Hiếu Thuận
- 21110728 - Nguyễn Dương Thế Vĩ

#### 1. NỘI DUNG KHI ĐỌC LƯỚI QUA BÀI BÁO

Bài báo trình bày một thuật toán sắp xếp các số dương tương tự thuật toán sắp xếp đã có trước đó (*Hoare's Quicksort*), nhưng thuật toán mang tính đơn hướng và không dùng đệ quy nên nó được tối ưu hơn, giúp chương trình hoạt động nhanh hơn và ngắn hơn. Thuật toán sắp xếp đó được gọi là “Thuật toán sắp xếp 1 chiều không chồng chéo” (*Stackles Quickssort Algorithm*).

#### 2. GẠCH CHÂN CÁC TỪ MỚI

- Link bài báo đã được gạch chân: [LINK](#)

#### 3. GIẢI THÍCH TỪ THEO TỪ ĐIỂN ANH-ANH

- Unidirectional: *having, moving in, or operating in only one direction*
- Recursion: *The process of repeating a function each time applying it to the result of the previous stage*
- Pivot: *Element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot*
- Destination: *the place where someone is going or where something is being sent or taken*
- Burn The Candle At Both Ends: *Trying to do two or more large projects at the same time.*
- Pointer: *A variable whose value is the address of another variable, i.e.,*
- Classical: *Traditional in style or form, or based on methods developed over a long period of time, and considered to be of lasting value.*
- Invariant: *not changing*
- Permutation: *Any of the different ways in which a set of things can be ordered*
- Vacan: *Empty.*
- Scheme : *A plan or system for doing or organizing something*
- Sentinel: *A soldier whose job is to guard something*
- Hypothetical: *Imagined or suggested but not necessarily real or true*

#### 4. GIẢI THÍCH TỪ THEO TỪ ĐIỂN ANH-VIỆT

- Unidirectional: *Hướng duy nhất.*
- Recursion: *Đệ quy là phương pháp dùng hàm để gọi lại chính nó để tiếp tục giải quyết trên dữ liệu đã khai báo trước đó.*
- Povit: *"Phần tử chốt" là phần chia mảng thành hai mảng con, khi sắp xếp các phần tử của hai mảng con sẽ lần lượt so sánh với "phần tử chốt", ta sẽ được một mảng gồm những phần tử nhỏ hơn hoặc bằng phần tử chốt và một mảng gồm những phần tử lớn hơn phần tử chốt.*
- Destination: *Nơi đến*
- Burn The Candle At Both Ends: *Làm việc gì đó quá sức, vắt kiệt sức mình.*
- Pointer: *Con trỏ.*
- Classical: *Kinh điển, cổ điển.*
- Invariant: *Không thay đổi.*
- Permutation: *Hoán vị.*
- Scheme :*Hệ thống tổ chức*
- Sentinel: *Lính canh (giám sát).*
- Hypothetical: *Giả thuyết.*

#### 5. LINK VIDEO, WEBSITE GIẢI THÍCH CHO CÁC KHÁI NIỆM:

+**Stack**: *is a linear data structure that follows the principle of Last In First Out (LIFO). This means the last element inserted inside the stack is removed first.*

- Link1: <https://www.programiz.com/dsa/stack>
- Link2: <https://www.geeksforgeeks.org/introduction-to-stack-data-structure-a/>

+**Quicksort**: *is a Divide and Conquer algorithm. It picks an element as a pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways.*

- Link1: <https://www.geeksforgeeks.org/quick-sort/>
- Link2: <https://en.wikipedia.org/wiki/Quicksort>

+**Hoare's Quicksort** : *works by initializing two indexes that start at two ends, the two indexes move toward each other until an inversion is (A smaller value on the left side and greater value on the right side) found. When an inversion is found, two values are swapped and the process is repeated.*

- Link1: [Geeksforgeeks](#)
- Link2: <https://www.youtube.com/watch?v=NuQYFXmLUrM>

+**Kirchoff's law** : *describe current in a node and voltage around a loop. These two laws are the foundation of advanced circuit analysis*

- Link1: [https://en.wikipedia.org/wiki/Kirchhoff%27s\\_circuit\\_laws](https://en.wikipedia.org/wiki/Kirchhoff%27s_circuit_laws)
- Link2: <https://www.youtube.com/watch?v=a2tmHmIQ3hw>

## 6. Ý NGHĨA TỪNG CÂU TRONG BÀI BÁO:

*Ghi chú này mô tả kỹ thuật sắp xếp tương tự như thuật toán "Quicksort" nổi tiếng, nhưng nó mang tính đơn hướng và không dùng đệ quy. Cách tiếp cận mới, giả định rằng các key được sắp xếp là các số dương, dẫn đến chương trình ngắn hơn nhiều.*

Hoare's "Quicksort" trong nhiều năm đã là phương pháp được lựa chọn cho mục đích sắp xếp, vì vòng lặp bên trong nhanh nên thời gian chạy trung bình hiệu quả trên hầu hết các máy. Cho một mảng có  $N$  records  $R_1...R_N$ , trong đó các records bao gồm các key  $K_1...K_N$ , chúng ta muốn sắp xếp lại sao cho các key theo thứ tự tăng dần. Ý tưởng chính của Quicksort là chọn một record "pivot"  $R$  và sắp xếp lại các record để  $R$  di chuyển đến đích chính xác của nó là  $R_i$ ; sau đó quy trình tương tự được áp dụng đệ quy cho các mảng con  $R_1...R_{i-1}$  và  $R_{i+1}...R_N$ .

Nếu tất cả các khóa  $K_j$  đều dương, chúng ta có thể cho key của record cuối cùng mang giá trị âm. Ngoài ra, còn có một cách để thực hiện thao tác xoay vòng cơ bản với các con trỏ từ trái sang phải trong mảng, thay vì "burning the candle at both ends" như trong phương pháp Quicksort cổ điển. Sự kết hợp của hai ý tưởng này dẫn đến một thuật toán không ngăn xếp (stackless algorithm) đơn giản đáng ngạc nhiên chỉ chạy chậm hơn khoảng 60% so với các triển khai nhanh nhất của Quicksort.

## 7. Ý NGHĨA TỪNG ĐOẠN TRONG BÀI BÁO:

- **Ý nghĩa đoạn 1:** Giới thiệu một thuật toán sắp xếp tương tự nhưng thuật toán "Quicksort" mà chúng ta đã biết trước đó, nhưng nó không dùng đệ quy nên làm cho chương trình ngắn hơn.
- **Ý nghĩa đoạn 2:** Mô tả cách hoạt động của thuật toán *Hoare's "Quicksort"* là một thuật toán sắp xếp đã được dùng từ nhiều năm trước. Thuật toán dùng mảng các records  $R_1...R_N$ , mảng các key  $K_1...K_N$  và phần tử "**pivot**" để thực hiện sắp xếp.
- **Ý nghĩa đoạn 3:** Sử dụng một key ở vị trí cuối cùng là số âm thì thuật toán sẽ tối ưu hơn. Sự kết hợp đó sẽ tạo thành một thuật toán sắp xếp mới hoạt động tối ưu hơn so với các thuật toán Quicksort cổ điển.
- **Ý nghĩa đoạn 4:** Mô tả và giải thích các bước để thực hiện thuật toán sắp xếp mới "*Stackless Quicksort*".
- **Ý nghĩa đoạn 5:** Nêu các bước cụ thể để thực hiện thuật toán và code chạy chương trình.
- **Ý nghĩa đoạn 6:** Dùng định luật Kirchhoff để tính độ phức tạp  $O$  của thuật toán mới. Ta thấy được nó tối ưu và hoạt động tốt hơn so với thuật toán Quicksort ban đầu.

## 8. PHÂN TÍCH CHI TIẾT CÔNG THỨC

**\*Xác định công thức tính thời gian chạy - độ phức tạp của thuật toán:**

- Sử dụng định luật Kirchhoff và vì có đúng một key là số âm ta có thể dễ dàng tính được thời gian chạy chương trình sẽ là:  $4C + 11X + 19N + 2$ .

Trong đó:

+  $C$  là số lần so sánh.

+  $X$  là số bước sắp xếp lại.

- Giá trị trung bình  $C_N$  và  $X_N$  của  $C$  và  $X$  thỏa mãn:

$$C_N = \frac{1}{N} \sum_{1 \leq i \leq N} (N + C_{i-1} + C_{N-i}),$$
$$X_N = \frac{1}{N} \sum_{1 \leq i \leq N} (i - 1 + X_{i-1} + X_{N-i})$$

- Khi  $N > 0$ ;  $C_0 = X_0 = 0$ . Nên:

$$C_N = 2(N + 1)H_N - 3N; \quad X_N = (N + 1)H_N - 2N.$$

- Do đó thời gian trung bình chạy chương trình là

$$19(N + 1)H_N - 15N + 2 \approx 19N \ln N - 4.033N + O(\log N).$$

————— **HẾT** —————