

**BÁO CÁO THỰC HÀNH MÔN PHÂN TÍCH THUẬT TOÁN****Lab 7 - Tuần 11****Đinh Anh Huy - 18110103**

**Bài toán 1.** Một mảng  $A$  có  $n$  phần tử. Ta gọi  $x \in A$  là phần tử "lần chiếm cấp độ 10" nếu như  $x$  xuất hiện ít nhất  $\left\lceil \frac{n}{10} \right\rceil$  lần trong  $A$ .

1. Xây dựng một thuật toán tìm phần tử lần chiếm cấp 10 trong một mảng.
2. Liệu có tồn tại thuật toán có độ phức tạp  $O(n)$  để tìm phần tử kiểu này không?

**Lời giải**

Lấy ý tưởng từ việc sử dụng *dictionary* để lưu trữ số lần xuất hiện của từng phần tử riêng biệt trong mảng  $A$ , ta xây dựng được thuật toán tìm phần tử lần chiếm cấp độ 10 trong  $A$  như sau

---

**Algorithm 1:** Thuật toán tìm phần tử lần chiếm cấp 10 trong một mảng.

---

**Function** Lan-Chiem-Cap-10( $A$ ):

```
 $n \leftarrow \text{len}(A);$ 
 $\text{counts} \leftarrow \text{empty dictionary};$ 
for  $i = 0 \rightarrow n - 1$  do
    if  $A[i] \in \text{counts.keys}$  then
         $\text{counts}[A[i]] \leftarrow \text{counts}[A[i]] + 1;$ 
    end
    else
         $\text{counts}[A[i]] \leftarrow 1;$ 
    end
end
 $\text{result} \leftarrow \text{empty array};$ 
for  $\text{value} \in \text{counts.keys}$  do
    if  $\text{counts}[\text{value}] \geq n//10$  then
         $\text{result} \text{ add } \text{value};$ 
    end
end
return  $\text{result};$ 
```

---

Thực thi thuật toán trên với mảng  $A$  được tạo ngẫu nhiên, cho kết quả như sau

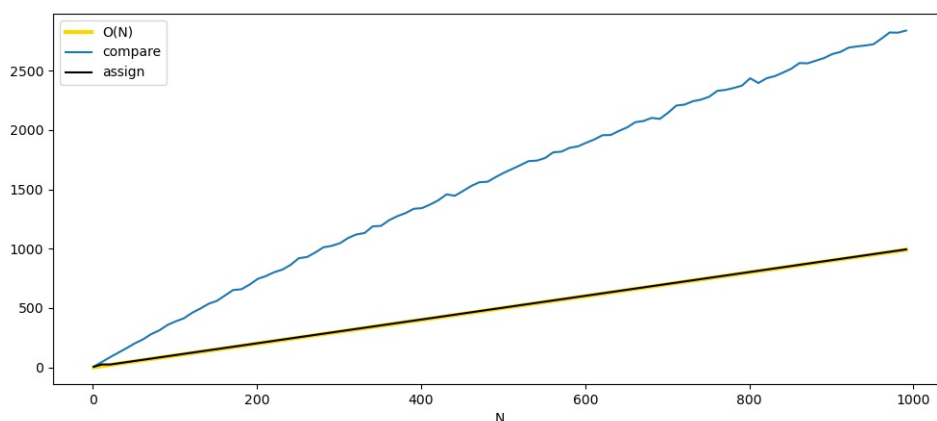
## Exercise 1

\*\*\*\*\*

Array A: [9 4 9 9 1 6 4 6 8 7 1 5 8 9 2 7 3 3 2 4]

Cac phan tu lan chiem cap do 10 trong A: [9, 4, 1, 6, 8, 7, 2, 3]

Từ thuật toán được trình bày ở trên, ta dễ dàng thấy rằng thuật toán này có độ phức tạp là  $O(N)$ . Để có cái nhìn trực quan hơn, ta sẽ xem xét biểu đồ các đường biểu diễn số phép so sánh và phép gán của thuật toán với đường  $O(N)$  như sau



Ta thấy rằng, đường biểu diễn số phép gán trùng với đường  $O(N)$  và đường biểu diễn số phép so sánh cũng có dạng đường thẳng và bị lệch lên so với đường  $O(N)$ . Do đó thuật toán này có độ phức tạp là  $O(N)$ . Đồng thời ý này cũng trả lời cho câu hỏi ở ý thứ 2 rằng có tồn tại thuật toán có độ phức tạp là  $O(N)$  thoả mãn đề bài.

**Bài toán 2.** Cho mảng  $A$  và  $B$  đã được sắp xếp theo thứ tự tăng dần và có  $n$  phần tử. Hãy xây dựng 1 thuật toán khả thi để merge hai mảng  $A$  và  $B$  thành mảng có  $2n$  phần tử đã được sắp xếp theo thứ tự tăng dần.

**Lời giải**

Sử dụng ý tưởng merge hai mảng đã được sắp xếp thành mảng có thứ tự tăng dần trong thuật toán *Merge Sort*. Các bước thực hiện như sau

1. Tạo một mảng con rỗng  $C$ .
2. Ta sẽ lấy ra phần tử nhỏ hơn trong 2 phần tử đang xét của 2 mảng  $A$  và  $B$  để thêm vào  $C$ . Thực hiện đến khi duyệt hết 1 trong 2 mảng  $A$  và  $B$ .
3. Thêm các phần tử còn lại trong mảng mà chưa duyệt hết ở bước 2 vào mảng  $C$ .

**Algorithm 2:** Thuật toán merge hai mảng có thứ tự.**Function Merge-Sorted-Arrays( $A, B$ ):**

```

 $n \leftarrow \text{len}(A)$ ;
 $C \leftarrow$  empty array;
 $i \leftarrow 0$ ;
 $j \leftarrow 0$ ;
while  $i < n$  and  $j < n$  do
    if  $A[i] < B[j]$  then
         $C \text{ add } A[i]$ ;
         $i \leftarrow i + 1$ ;
    end
    else
         $C \text{ add } B[j]$ ;
         $j \leftarrow j + 1$ ;
    end
end
while  $i < n$  do
     $C \text{ add } A[i]$ ;
     $i \leftarrow i + 1$ ;
end
while  $j < n$  do
     $C \text{ add } B[j]$ ;
     $j \leftarrow j + 1$ ;
end
return  $C$ ;

```

Thực thi thuật toán trên với 2 mảng  $A, B$  được tạo ngẫu nhiên cho kết quả như sau

**Exercise 2**

```
*****
```

```
A sorted A = [1 2 4 6 9]
```

```
A sorted B = [2 2 5 6 7]
```

```
Merged array = [1, 2, 2, 2, 4, 5, 6, 6, 7, 9]
```

Ta thấy rằng thuật toán trên cũng có độ phức tạp là  $O(N)$ , ta dễ dàng thấy qua biểu đồ trực quan biểu diễn số phép gán và so sánh của thuật toán trên và đường  $O(N)$  như sau

