

Phân Tích Thuật Toán

Bùi Thị Thanh Phương

Ngày 23 tháng 5 năm 2023

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 $\lfloor \frac{n}{10} \rfloor$ 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

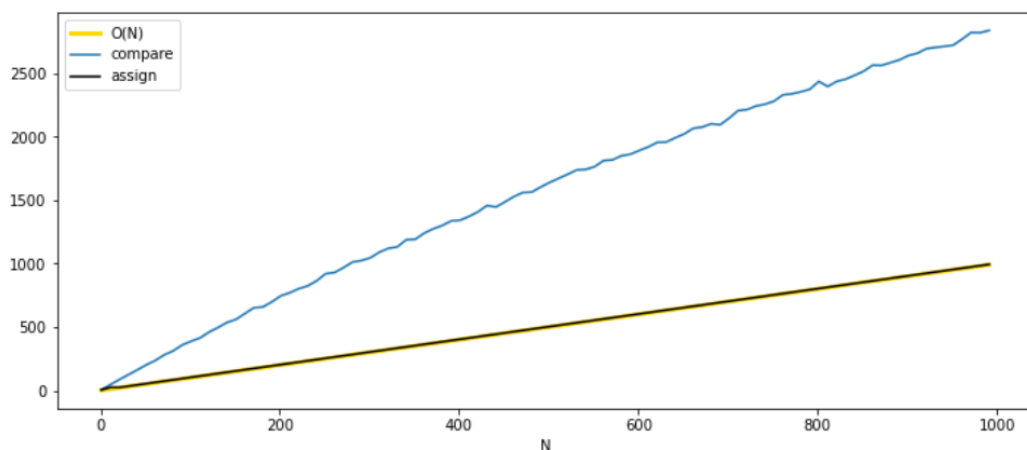
```
1: function LAN-CHIEM-CAP-10( $A$ )
2:    $n \leftarrow \text{len}(A)$ 
3:    $\text{counts} \leftarrow$  empty dictionary
4:   for  $i = 0 \rightarrow n - 1$  do
5:     if  $A[i] \in \text{counts.keys}$  then
6:        $\text{counts}[A[i]] \leftarrow \text{counts}[A[i]] + 1$ 
7:     else
8:        $\text{counts}[A[i]] \leftarrow 1$ 
9:     end if
10:  end for
11:   $\text{result} \leftarrow$  empty array
12:  for  $\text{value} \in \text{counts.keys}$  do
13:    if  $\text{counts}[\text{value}] \geq n/10$  then
14:       $\text{result}$  add  $\text{value}$ 
15:    end if
16:  end for
17:  return  $\text{result}$ 
18: end function
```

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]
Các phần tử lần chiếm cấp độ 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 chệch 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 .
- Thực thi thuật toán trên với 2 mảng A, B được tạo ngẫu nhiên cho kết quả như sau

IV

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

Algorithm 2 Thuật toán merge hai mảng có thứ tự

```

1: function MERGE-SORTED-ARRAYS( $A, B$ )
2:    $n \leftarrow \text{len}(A)$ 
3:    $C \leftarrow$  empty array
4:    $i \leftarrow 0$ 
5:    $j \leftarrow 0$ 
6:   while  $i < n$  and  $j < n$  do
7:     if  $A[i] < B[j]$  then
8:        $C \text{ add } A[i]$ 
9:        $i \leftarrow i + 1$ 
10:    else
11:       $C \text{ add } B[j]$ 
12:       $j \leftarrow j + 1$ 
13:    end if
14:  end while
15:  while  $i < n$  do
16:     $C \text{ add } A[i]$ 
17:     $i \leftarrow i + 1$ 
18:  end while
19:  while  $j < n$  do
20:     $C \text{ add } B[j]$ 
21:     $j \leftarrow j + 1$ 
22:  end while
23:  return  $C$ 
24: end function

```

