

Phân tích độ phức tạp thuật toán không đệ quy

NGUYỄN TUẤN ANH, NGUYỄN TRẦN VIỆT ANH*

Ngày 22 tháng 3 năm 2023

Bài viết này trình bày lời giải cho các bài tập phân tích độ phức tạp thuật toán không đệ quy, do **nhóm 13** đề xuất.

Mục lục

1	Bài toán	2
2	Lời giải	3
2.1	Problem 10/60, proposed by Team 13	3
2.2	Problem 11/61, proposed by Team 13	4
2.3	Problem 11/69, proposed by Team 13	5

*Sinh viên lớp KHTN2021, MSSV: 21520142 - 21520006

§1 Bài toán

Các bài toán có nguồn từ [Introduction to The Design and Analysis of Algorithms](#).

1. The **range** of a finite nonempty set of n real number S is defined as the difference between the largest and smallest elements of S . For each representation of S given below, describe in English an algorithm to compute the range. Indicate the time efficiency classes of these algorithms using the most appropriate notation (O , Θ , or Ω).
 - a. An unsorted array.
 - b. A sorted array.
 - c. A sorted singly linked list.
 - d. A binary search tree.
2. *Lighter or heavier?* You have $n > 2$ identical-looking coins and a two-pan balance scale with no weights. One of the coins is a fake, but you do not know whether it is lighter or heavier than the genuine coins, which all weigh the same. Design a $\Theta(1)$ algorithm to determine whether the fake coin is lighter or heavier than the others.
3. Consider the following version of an important algorithm that we will study later in the book.

```
ALGORITHM  GE( $A[0..n-1, 0..n]$ )  
  //Input: An  $n \times (n+1)$  matrix  $A[0..n-1, 0..n]$  of real numbers  
  for  $i \leftarrow 0$  to  $n-2$  do  
    for  $j \leftarrow i+1$  to  $n-1$  do  
      for  $k \leftarrow i$  to  $n$  do  
         $A[j, k] \leftarrow A[j, k] - A[i, k] * A[j, i] / A[i, i]$ 
```

- a. Find the time efficiency class of this algorithm.
- b. What glaring inefficiency does this pseudocode contain and how can it be eliminated to speed the algorithm up?

§2 Lời giải

§2.1 Problem 10/60, proposed by Team 13

Bài toán

The **range** of a finite nonempty set of n real number S is defined as the difference between the largest and smallest elements of S . For each representation of S given below, describe in English an algorithm to compute the range. Indicate the time efficiency classes of these algorithms using the most appropriate notation (O , Θ , or Ω).

- An unsorted array.
- A sorted array.
- A sorted singly linked list.
- A binary search tree.

a. Với một mảng chưa được sắp xếp, ta không biết quá nhiều thông tin về các phần tử. Nếu ta chọn sắp xếp lại sẽ tốn chi phí $O(n \log n)$. Một cách khác đó là đi qua lần lượt tất cả phần tử với biến tạm để lưu giá trị nhỏ nhất, lớn nhất hiện thời, độ phức tạp thời gian là $O(n)$.

b. Khoảng cách giữa phần tử đầu và cuối của mảng chính là phạm vi. Độ phức tạp $O(1)$.

c. Dù là danh sách liên kết đã được sắp xếp nhưng chúng ta vẫn phải duyệt toàn bộ, tương tự như mảng chưa sắp xếp. Độ phức tạp $O(n)$.

d. Trong một cây nhị phân, khoảng cách giữa nút con trái và phải ngoài cùng chính là phạm vi. Việc tìm chúng sẽ mất $\Theta(\log n)$ trong trường hợp cây cân bằng, nếu không sẽ phải duyệt toàn bộ với độ phức tạp $O(n)$.

§2.2 Problem 11/61, proposed by Team 13

Bài toán

Lighter or heavier? You have $n > 2$ identical-looking coins and a two-pan balance scale with no weights. One of the coins is a fake, but you do not know whether it is lighter or heavier than the genuine coins, which all weigh the same. Design a $\Theta(1)$ algorithm to determine whether the fake coin is lighter or heavier than the others.

- 1) Nếu $n \equiv 0 \pmod{3}$. Chia tập n đồng xu thành 3 nhóm, sẽ có hai nhóm cùng trọng lượng và nhóm còn lại chứa đồng xu giả khác trọng lượng.
- 2) Nếu $n \equiv 1 \pmod{3}$. Lấy 3 đồng xu bất kỳ, cân và chọn ra hai đồng xu cùng trọng lượng. Sau đó loại bỏ một đồng xu. Lúc này quay lại trường hợp 1.
- 3) Nếu $n \equiv 2 \pmod{3}$. Ở trường hợp 2, ta loại bỏ hẳn hai đồng cùng trọng lượng bởi vì chúng chắc chắn là thật. Quay lại trường hợp 1.

Dù có bao nhiêu đồng xu thì cũng chỉ cần kiểm tra theo modulo 3 của n , sau đó tốn không quá 2 bước kiểm tra. Như vậy độ phức tạp của thuật toán là $\Theta(1)$.

§2.3 Problem 11/69, proposed by Team 13

Bài toán

Consider the following version of an important algorithm that we will study later in the book.

- Find the time efficiency class of this algorithm.
- What glaring inefficiency does this pseudocode contain and how can it be eliminated to speed the algorithm up?

ALGORITHM $GE(A[0..n-1, 0..n])$
 //Input: An $n \times (n+1)$ matrix $A[0..n-1, 0..n]$ of real numbers
for $i \leftarrow 0$ **to** $n-2$ **do**
 for $j \leftarrow i+1$ **to** $n-1$ **do**
for $k \leftarrow i$ **to** n **do**
 $A[j, k] \leftarrow A[j, k] - A[i, k] * A[j, i] / A[i, i]$

Hình 1: Pseudocode

a. Với mỗi i ở vòng lặp ngoài cùng, cả hai biến vòng lặp j, k chỉ chạy theo i . Hai vòng lặp trong mất $O(n^2)$, kết hợp cả 3 vòng lặp thì thuật toán có độ phức tạp $O(n^3)$.

b. Nhận thấy rằng nếu $k = i$ thì $A[j, i] \leftarrow 0 = A[j, i] - A[i, i] * A[j, i] / A[i, i]$.

Như vậy biểu thức tính toán có thể được viết lại $A[j, k] \leftarrow A[j, k]$.

Điều này là hiển nhiên! Vì vậy vòng lặp trong cùng không cần thiết.

Ta loại bỏ nó và sửa lại biểu thức thành $A[j, i] = 0$.

Độ phức tạp lúc này giảm xuống còn $O(n^2)$.