ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Lớp: CS112.Q11.KHTN

Môn học: Phân Tích và Thiết Kế Thuật Toán

Chủ đề 2 - Phân tích độ phức tạp thời gian của đệ quy

Nhóm 4:
Lê Văn Thức
Bảo Quý Định Tân

Giảng viên:
Nguyễn Thanh Sơn

BÀI 1

Giả sử ô vuông ban đầu ở vị trí có tọa độ là (0,0). Ta nhận thấy, ở thời gian N, các ô mới được tô thỏa $|x| + |y| = n$, có tổng cộng $4n$ ô thỏa mãn điều kiện này.

Gọi $f(n)$ là số ô được tô sau n thời gian, theo phân tích trên, ta có công thức truy hồi:

$$f(n) = \begin{cases} 1, & \text{nếu } n = 0, \\ f(n-1) + 4n, & \text{nếu } n \geq 1. \end{cases}$$

Công thức truy hồi thời gian $T(n)$ của công thức truy hồi trên là:

$$T(n) = \begin{cases} 1, & \text{nếu } n = 0, \\ T(n-1) + 1, & \text{nếu } n \geq 1. \end{cases}$$

$$\Leftrightarrow \begin{cases} 1, & \text{nếu } n = 0, \\ n + 1, & \text{nếu } n \geq 1. \end{cases}$$

Vậy độ phức tạp của thuật toán là: $\Theta(n)$

BÀI 2

2.a

- Với $n <= 2$, cần ít nhất 2 phút để nấu cả 2 mặt của chiếc bánh.
- Với $n > 2$, chọn 2 cái bánh để nấu trong 2 phút và giải quyết bài toán với $n - 2$ cái còn lại.

Vậy ta có công thức truy hồi:

$$f(n) = \begin{cases} 2, & \text{nếu } n \leq 2, \\ f(n-2) + 2, & \text{nếu } n \geq 3. \end{cases}$$

Công thức truy hồi thời gian của công thức trên là:

$$T(n) = \begin{cases} 1, & \text{nếu } n \leq 2, \\ T(n-2) + 1, & \text{nếu } n \geq 3. \end{cases}$$

$$\Leftrightarrow \begin{cases} 1, & \text{nếu } n \leq 2, \\ \left\lceil \frac{n}{2} \right\rceil, & \text{nếu } n \geq 1. \end{cases}$$

Vậy độ phức tạp của thuật toán là: $\Theta(n)$

2.b

Với n = 3, $f(3) = f(1) + 2 = 4$(phút), trong khi đó, ta có thể:

- Trong phút thứ nhất, nấu mặt trên của bánh 1 và bánh 2.
- Trong phút thứ hai, nấu mặt dưới của bánh 1 và mặt trên của bánh 3.
- Trong phút thứ ba, nấu mặt dưới của bánh 2 và mặt dưới của bánh 3.

Như vậy, chỉ cần 3 phút để nướng toán bộ bánh ⇒ thuật toán chưa trả về kết quả tối ưu.

Thật vậy, với mỗi giá trị $n \geq 3$ lẻ, việc dành 2 phút chỉ để nấu 1 bánh cuối cùng là không tối ưu.

2.c

Thay vì xử lý riêng lẻ một chiếc bánh ta coi một mặt bánh là đơn vị cần nấu. Có tổng cộng $2n$ mặt bánh cần được nấu, và mỗi lần ta có thể nấu tối đa 2 mặt trong 1 phút.

Do đó, ta có trường hợp cơ sở là $n \leq 2$ tốn 1 phút khi còn 2 mặt bánh. Ở các trường hợp khác, ta chọn 2 mặt bánh bất kỳ để nấu và giải bài toán với $n - 2$ mặt còn lại.

Công thức truy hồi là:

$$f(n) = \begin{cases} 1, & \text{nếu } n \leq 2, \\ f(n-2) + 1, & \text{nếu } n \geq 3. \end{cases}$$

Như vậy, với 3 chiếc bánh, ta có $n = 6$ ta chỉ mất $f(6) = 3$ phút, trùng với kết quả tối ưu trên và ngắn hơn $f(3) = 4$ của thuật toán ban đầu.

Chỉ riêng trường hợp 1 chiếc bánh, ta không thể áp dụng thuật toán này bởi ta chỉ không thể nấu 2 mặt của 1 chiếc bánh cùng lúc.

Trong các trường hợp còn lại, thuật toán tối ưu bởi nếu tồn tại n chiếc bánh và $2n$ mặt bánh thì thời gian tối thiểu để nướng bánh là $\frac{2n}{2} = n$(phút).

Ta lại có:

$$f(n) = \begin{cases} 1, & \text{nếu } n \leq 2, \\ f(n-2) + 1, & \text{nếu } n \geq 3. \end{cases}$$

$$\Leftrightarrow f(n) = \begin{cases} 1, & \text{nếu } n \leq 2, \\ \left\lceil \frac{2n}{2} \right\rceil, & \text{nếu } n \geq 3. \end{cases}$$

Mà tham số đầu vào của hàm số trên luôn chẵn vì số mặt bánh là 2n, ta thấy, kết quả của hàm số luôn bằng n, trùng với kết quả tối ưu(chỉ trừ riêng trường hợp $n = 1$) vậy, thuật toán trên là thuật toán hiệu quả nhất.

BÀI 3

1. Xác định tham số kích thước input

- Kích thước đầu vào: n

2. Xác định phép toán cơ bản của thuật toán

Phép toán cơ bản của thuật toán là phép nhân và phép cộng trong vòng lặp

3. Hành vi của thuật toán khi gặp input có cùng độ lớn

Với input giống nhau, số lần thực hiện các phép toán cơ bản thuật toán không thay đổi vì chỉ dựa trên duy nhất 1 tham số n.

4. Thiết lập hàm đệ quy

$$f(n) = \begin{cases} 1, & \text{nếu } n \leq 1, \\ \sum_{i=0}^{n-1} f(i) + f(n-i-1), & \text{nếu } n \geq 2. \end{cases}$$

5. Giải hệ thức truy hồi

Gọi T(n) là thời gian chạy của hàm, xét n>=2 ta có:

$$T(n) = \sum_{i=0}^{n-1} T(i) + T(n-i-1)$$

$$= 2\sum_{i=0}^{n-1} [T(i)]$$

$$= 2\sum_{i=0}^{n-1} T(i)$$

Đặt:

$$S(n) = \sum_{i=0}^{n} T(i)$$

Khi đó, ta có:

$$T(n) = 2S(n-1)$$

Suy ra:

$$S(n) = S(n-1) + T(n) = S(n-1) + 2S(n-1) = 3S(n-1)$$

Giải hệ thức truy hồi cho $S(n)$, ta được:

$$S(n) = 3^n S(0)$$

Suy ra:

$$T(n) = 2S(n-1) = 2 \cdot 3^{n-1} S(0)$$

Do đó:

$$T(n) = O(3^n)$$

Vậy độ phức tạp của đoạn code trên là $O(3^n)$