5.9 Phân tích thuật toán

Ta đếm các phép toán trong thuật toán, gồm phép toán số học, phép so sánh, phép toán logic, phép gán rồi đánh giá độ phức tạp theo cỡ của dữ liệu đầu vào.

Ví dụ 5.67. An mở tài khoản 100 triệu ở ngân hàng với lãi suất 5‰/tháng. Đầu chu kỳ mỗi tháng, An gửi thêm 50 triệu.

a) Xác định số dư trong tài khoản của An sau 1 tháng, 2 tháng.

b) Mô tả thuật toán xác định số dư sau n tháng.

c) Xác định độ phức tạp của thuật toán ở ý (b).

Giải. a)

n	Số dư sau n tháng (đơn vị: triệu)
1	$100 + 100 \cdot 0.005 + 50 = 150.5$
2	$150.5 + 150.5 \cdot 0.005 + 50 = 201.253$

b) Thuật toán tìm số dư tài khoản sau n tháng.

Data: n

Result: Số dư tài khoản sau n tháng

```
1 so_du = 100                                   // khởi tạo số dư
2 gui_them = 50                                 // tiền gửi thêm hàng tháng
3 lai_suat = 0.005                              // lãi suất tháng
4 i = 1                                         // khởi tạo biến đếm
5 while i ≤ n do
6     so_du = so_du + so_du * lai_suat + gui_them
7     i = i + 1
```

c) 1) Dòng 1, 2, 3 có 3 phép gán.

2) Dòng 4, 5, 7 cho n vòng lặp. Dòng 4 có 1 phép gán, dòng 5 có $n + 1$ phép so sánh, ứng với $i = \overline{1, n+1}$, dòng 7 có 1 phép cộng và 1 phép gán (lặp n lần). Ba dòng này có số phép toán $1 + (n + 1) + 2n = 3n + 2$, với n là số vòng lặp.

3) Dòng 6 có 3 phép toán số học và 1 phép gán (lặp n lần).

Vậy, số phép toán của thuật toán là $f(n) = 3 + (3n + 2) + 4n = 7n + 5 \in O(n)$.

□

Khi n lớn, "cấp độ lớn" của $7n + 5$ chủ yếu phụ thuộc vào n, là số lần lặp của vòng lặp **while**. Do đó để chỉ ra $f \in O(n)$, ta chỉ cần đếm số chu trình của vòng lặp **while**.

Các ví dụ về độ phức độ phức tạp (trong trường hợp) tốt nhất, độ phức tạp xấu nhất và độ phức tạp trung bình.

Ví dụ 5.68 (Thuật toán tìm kiếm tuyến tính)**.** Mô tả thuật toán và xác định độ phức tạp của thuật toán tìm vị trí của x trong dãy $a_1, a_2, \dots, a_n$.

Giải. a)

Data: $a_1, a_2, \dots, a_n$ và x
Result: Vị trí của x trong dãy

```
1 i = 1                                                    // chỉ số dãy
2 while i ≤ n and x ≠ a_i do i = i + 1
3 if i ≤ n then
4 |  viTri = i                                            // tìm được
5 viTri = 0                                              // không tìm được
// viTri là chỉ số của phần tử đầu tiên bằng x, là 0 nếu không tìm được
```

b) Đặt $f(n)$ là số chu trình thực hiện trong vòng lặp **while** ở dòng 2.

i) Nếu $x = a_1$, thì $f(n) = 1 \in O(1)$, là độ phức tạp tốt nhất.

ii) Nếu $x \neq a_i \ \forall i = \overline{1, n}$, hoặc dãy $a_1, a_2, \dots, a_n$ phân biệt và $x = a_n$, thì $f(n) = n \in O(n)$, là độ phức tạp xấu nhất.

iii) Giả sử $P(x \in \{a_1, a_2, \dots, a_n\}) = p$, trong đó x nhận mỗi giá trị với khả năng như nhau $P(x = a_i) = \dfrac{p}{n}$ và $P(x \in \{a_1, a_2, \dots, a_n\}) = 1 - p$.
Đặt $X = f(n)$. Ta có

$$P(X = i) = P(x = a_i) = \frac{p}{n}, \ \forall i = \overline{1, n-1}, \quad \text{và}$$
$$P(X = n) = P(x = a_n) + P(X \notin \{a_1, a_2, \dots, a_n\}) = \frac{p}{n} + (1 - p)$$

Khi đó

$$\begin{aligned} EX &= \sum_{i=1}^{n} i \cdot P(X = i) = 1 \cdot \frac{p}{n} + 2 \cdot \frac{p}{n} + \cdots + n\Big(\frac{p}{n} + 1 - p\Big) \\ &= (1 + 2 + \cdots + n)\frac{p}{n} + n(1 - p) = \frac{n(n+1)}{2} \cdot \frac{p}{n} + n(1 - p) \\ &= \frac{(n+1)p}{2} + n(1 - p) = n(1 - \frac{p}{2}) + \frac{p}{2} \in O(n) \end{aligned}$$

là độ phức tạp trung bình của thuật toán.

□

Ví dụ 5.69. Trình bày các thuật toán tính a^n với $a \in \mathbb{R}$, $n \in \mathbb{N}$, và đánh giá độ phức tạp tương ứng.

Giải. **Thuật toán 1:**

Data: a, n
Result: a^n
1 $x = 1$ // x lưu $a^1, a^2, \ldots, a^n$
2 **for** $i = 1$ **to** n **do**
3 $\quad x = x * a$

Vòng lặp **for** ở dòng 2 thực hiện n chu trình, nên thuật toán có độ phức tạp $f(n) = n \in O(n)$, là độ phức tạp tuyến tính.

Thuật toán 2: phương pháp chia đôi để tính lũy thừa

1 $x = 1$
2 $i = n$
3 **while** $i > 0$ **do**
4 $\quad$ **if** i *lẻ* **then** $x = x * a$
5 $\quad a = a * a$
6 $\quad i = \lfloor \frac{i}{2} \rfloor$

Chẳng hạn, tính a^{10}

Bước	x	a	i
Khởi tạo	1	a	10
Dòng 3, chu trình 1	1	a^2	**5**
Dòng 3, chu trình 2	a^2	a^4	2
Dòng 3, chu trình 3	a^2	a^8	**1**
Dòng 3, chu trình 4	a^{10}	a^{16}	$\boxed{0}$

Đặt $f(n)$ là số chu trình của của vòng lặp **while**. Ta chứng minh $f(n) \leq 1+\log_2 n,\ \forall n \in \mathbb{Z}^+$ bằng quy nạp.

1) $f(1) = 1 \leq 1 + \log_2 1$: đúng.

2) Giả sử $f(k) \leq 1 + \log_2 k,\ \forall k = \overline{1, n}$. Ta sẽ chứng minh $f(n+1) \leq 1 + \log_2(n+1)$. Xét thuật toán với đầu vào $n+1$. Sau bước đầu tiên, $i = \lfloor \frac{n+1}{2} \rfloor$, từ đó $f(n+1) = 1 + f\left(\lfloor \frac{n+1}{2} \rfloor\right)$. Mặt khác, $1 \leq \frac{n+1}{2} \leq n$ nên $1 \leq \lfloor \frac{n+1}{2} \rfloor \leq n$, suy ra

$$f\left(\lfloor \frac{n+1}{2} \rfloor\right) \leq 1 + \log_2 \lfloor \frac{n+1}{2} \rfloor \leq 1 + \log_2 \frac{n+1}{2} = \log_2 (n+1).$$

Do đó $f(n+1) \leq 1 + \log_2 (n+1)$.

Vậy $f(n) \leq 1 + \log_2 n \in O(\log_2 n)$, thuật toán có độ phức tạp loga.

□

Cỡ đầu vào n	Độ phức tạp					
	$\log_2 n$	n	$n \log_2 n$	n^2	2^n	$n!$
2	1	2	2	4	4	2
16	4	16	64	256	6.5×10^4	2.1×13
64	6	64	384	4096	1.84×10^{19}	$> 10^{89}$

Cho tập A cỡ n. Xét hai thuật toán:

1) Tìm các tập con cỡ 1 của A. Có n tập con.

2) Tìm tất cả tập con của A. Có 2^n tập con.

Giả sử máy tính xác định mỗi tập con của A mất một nano giây (10^{-9}s). Khi đó nếu $|A| = 64$, thuật toán (1) thực thi xong gần như ngay lập tức, trong 64 nano giây. Tuy nhiên, thuật toán (2) thực hiện trong

$$1.84 \times 10^{19} \text{ nano giây } = 2.14 \times 10^5 \text{ ngày } = 585 \text{ năm.}$$

Bài tập bổ sung

5.38. Ước tính mất bao lâu để phân tích nguyên tố cho một số có 1000 chữ số bằng phép chia thử. Giả sử rằng ta thử tất cả các ước có thể đến căn bậc hai của số đó và có thể thực hiện một triệu tỷ phép chia thử mỗi giây (cỡ siêu máy tính). Chọn một đơn vị thời gian hợp lý cho câu trả lời.