

Bài tập chương 1

Phần 3

A. Thuật toán đệ quy

Bài 1. Cho hàm đệ quy sau

$$a) f(n) = \begin{cases} 2 & \text{nếu } n \leq 2 \\ 3f(n-2) + f(n-1) + 4 & \text{nếu } n > 2 \end{cases}$$

Hãy tính $f(4)$, $f(9)$

$$b) f(n) = \begin{cases} 1 & \text{nếu } n \leq 0 \\ f(n/2) + f(n-1) & \text{nếu } n > 0 \end{cases}$$

Hãy tính $f(3)$, $f(8)$

$$c) f(n) = \begin{cases} 1 & \text{nếu } n = 0 \\ 2 & \text{nếu } n = 1 \\ 2f(n\%2) + f(n/2) & \text{nếu } n > 1 \end{cases}$$

Hãy tính $f(4)$, $f(7)$

$$d) f(n) = \begin{cases} 1 & \text{nếu } n \leq 0 \\ 2f(n/2) + 1 & \text{nếu } n > 0 \text{ và chẵn} \\ 2f(n-1) & \text{nếu } 0 < n \text{ và lẻ} \end{cases}$$

Hãy tính $f(4)$, $f(7)$

B. Phương pháp thế

Bài 1. Chứng minh rằng

- a) $T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1$ là $O(\log n)$
- b) $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ là $O(n \log n)$
- c) $T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor + 12\right) + 1$ là $O(\log n)$
- d) $T(n) = 4T\left(\frac{n}{2}\right) + n$ là $O(n^2)$
- e) $T(n) = 2T\left(\frac{n}{2}\right) + 1$ là $O(n)$

Bài 2. Giải công thức đệ quy

a) $T(n) = 2T(\sqrt{n}) + 1$

Bài 3. Giải công thức đệ quy của thuật toán sắp xếp trộn – mergeSort

$$T(n) = \begin{cases} O(1) & \text{nếu } n = 1 \\ 2T\left(\frac{n}{2}\right) + O(n) & \text{nếu } n > 1 \end{cases}$$

C. Cây đệ quy

Bài 1. Xác định một cận trên tốt cho công thức đệ quy $T(n) = 3T\left(\frac{n}{2}\right) + n$ dùng phương pháp thế để xác nhận lại kết quả.

Bài 2. Vẽ cây đệ quy cho $T(n) = 4T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + cn$ với c là hằng số. Đưa ra tiệm cận chặt cho công thức đệ quy trên. Xác nhận lại lời giải bằng phương pháp thế.

D. Định lý thợ

Bài 1. Dùng định lý thợ để đưa ra các tiệm cận chặt cho các công thức đệ quy sau

- a) $T(n) = 4T\left(\frac{n}{2}\right) + n$
- b) $T(n) = 4T\left(\frac{n}{2}\right) + n^2$
- c) $T(n) = 4T\left(\frac{n}{2}\right) + n^3$

Bài 2. Dùng định lý thợ để chứng minh thời gian thực hiện của thuật toán tìm kiếm nhị phân $T(n) = T\left(\frac{n}{2}\right) + \Theta(1)$ là $\Theta(\log n)$

Bài 3. Định lý thợ có thể áp dụng cho công thức đệ quy $T(n) = 4T\left(\frac{n}{2}\right) + n^2 \log n$ được không, Tại sao? Tìm tiệm cận giới hạn trên cho $T(n)$

Bài 4. Đánh giá thời gian thực hiện của thuật toán đệ quy sau theo mô hình O-lớn

```
int findMin(int S[], int start, int end)
{
    if(start >= end) return S[end];

    int div = (end - start) / 2;
    int A, B;

    A = findMin(S, start, start + div);
    B = findMin(S, start + div + 1, end);

    return Min(A, B);
}
```

Trong đó hàm Min(A,B) trả về giá trị nhỏ nhất trong 2 số A, B và thời gian thực hiện là $\Theta(1)$

E. Viết chương trình

Bài 1. Hãy viết hàm `gcd()` để tìm ước số chung lớn nhất của 2 số nguyên dương a, b theo các thuật toán sau:

- a) **Phương pháp vét cạn:** duyệt tất cả các số nguyên dương có thể cho tới khi tìm thấy
- b) **Dùng thuật toán Euclid:** giả sử b là số nhỏ hơn, nếu $b = 0$ thì ước số chung lớn nhất là a , ngược lại thì ước số chung lớn nhất của a và b cũng là ước số chung lớn nhất của b và $a \% b$ (chia module, chia lấy phần dư). Cài đặt thuật toán Euclid theo 2 cách: lặp và đệ quy

Bài 2. Hãy viết chương trình để in ra màn hình tam giác số Pascal dạng hình vuông như sau

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

$$C(n, 0) = C(n, n) = 1 \text{ với } n \geq 0$$

$$C(n, k) = C(n - 1, k) + C(n - 1, k - 1) \text{ với } n > k > 0$$

- a) Vẽ cây đệ quy để tính $C(6, 4)$
- b) Xây dựng hàm đệ quy để tính $C(n, k)$
- c) Vẽ tam giác Pascal dạng hình vuông
- d) Cải tiến hàm tính $C(n, k)$ theo 2 cách: dùng đệ quy có nhớ, và không dùng đệ quy (dùng phương pháp lặp)
- e) Đánh giá độ phức tạp của hàm $C(n, k)$ trong các trường hợp c, d theo các tiêu chí: bộ nhớ, thời gian

Bài 3. Hàm Ackermann được định nghĩa như sau

$$A(0, n) = n + 1 \text{ với } n > 0$$

$$A(m, 0) = A(m - 1, 1) \text{ với } m > 0$$

$$A(m, n) = A(m - 1, A(m, n - 1)) \text{ với } m > 0 \text{ và } n > 0$$

- a) Tính toán các giá trị hàm Ackermann trong các trường hợp sau

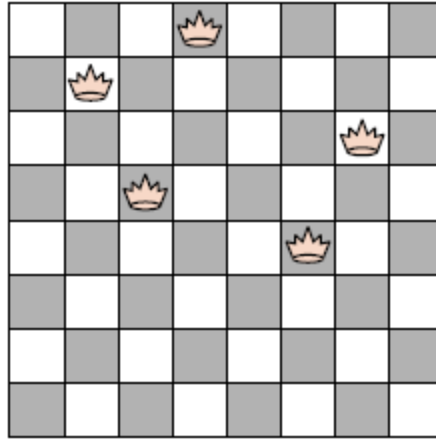
$$A(0, 9) \quad A(0, 9) \quad A(1, 8) \quad A(2, 2) \quad A(2, 0) \quad A(2, 3) \quad A(3, 2) \quad A(4, 2) \quad A(4, 3) \quad A(4, 0)$$

- b) Viết hàm đệ quy để tính giá trị hàm Ackermann
- c) Viết hàm không đệ quy để tính giá trị hàm Ackermann

F. Thuật toán quay lui

Bài 1. Chiều cao tối đa của cây đệ quy của hàm **solve_from** trong bài toán 8 hậu ?

Bài 2. Thực hiện tiếp hàm **solve_from** để giải bài toán 8 hậu với trạng thái hiện tại của bàn cờ là



Chú ý : trong trường hợp này ta xếp các quân hậu lần lượt theo hàng chứ không phải theo cột

Bài 3. Thực hiện thuật toán backtracking bằng tay để tìm ra một lời giải cho bài toán đặt 5 quân hậu trên bàn cờ kích thước 5x5

Bài 4. Cài đặt thuật toán backtracking để in ra lời giải có thể của bài toán 8 hậu

Bài 5. Xây dựng thuật toán backtracking để in ra tất cả các hoán vị của dãy {A,B,C,D,E}

Bài 6. Xây dựng thuật toán backtracking để giải bài toán ma phương (bậc chẵn và bậc lẻ)

Tham khảo : <http://mathworld.wolfram.com/MagicSquare.html>