LAB 3: Phân tích thuật toán (Tiếp theo)

1 Master theorem

1.1

Sử dụng Master Theorem để giải các phương trình đệ quy sau:

a. $T(n) = 9T(\frac{n}{3}) + n$

$$\begin{aligned} & a = 9; \quad b = 3; \quad f(n) = n \\ & n^{\log_b a} = n^{\log_3 9} = n^2 \\ & f(n) = O(n^{\log_b a - 1}) = n^1, \qquad \epsilon = 1 \\ \Rightarrow & f(n) = \Theta(n^{\log_b a}) = \Theta(n^2) \end{aligned}$$

b. $T(n) = T(\frac{2n}{3}) + 1$

$$\begin{aligned} & a = 1; \quad b = 3/2; \quad f(n) = 1 \\ & n^{\log_b a} = n^{\log_{3/2} 1} = 1 \\ & f(n) = O(1) \\ \Rightarrow & f(n) = \Theta(\log n) \end{aligned}$$

c. $T(n) = 3T(\frac{n}{4}) + n \log n$

$$\begin{aligned} & a = 3; \quad b = 4; \quad f(n) = n \log n \\ & n^{\log_b a} = n^{\log_4 3} = n^{0.79} \\ & f(n) = n \log n = \Omega(n^{\log_4 3 + \epsilon}) \\ & af(\frac{n}{b}) = \frac{3n}{4} \log(\frac{n}{4}) = \frac{3n}{4} \log n + \frac{3n}{4} \log 4 \leq c \log n, \quad \exists c = \frac{3}{4} < 1 \\ \Rightarrow & f(n) = \Theta(n \log n) \end{aligned}$$

d. $T(n) = 2T(\frac{n}{3}) + n$

$$
\begin{aligned}
& a = 2; \quad b = 3; \quad f(n) = n \\
& n^{\log_b a} = n^{\log_3 2} = n^0.63 \\
& f(n) = n = \Omega(n^{0.63+0.37}), \qquad \epsilon = 0.37 > 0 \\
& af(\frac{n}{b}) = 2f(\frac{n}{3}) = 2\frac{n}{3} \leq cn, \qquad c = \frac{2}{3} < 1 \\
\Rightarrow & f(n) = \Theta(n)
\end{aligned}
$$

e. $T(n) = T(\frac{n}{2}) + n$

$$
\begin{aligned}
& a = 1; \quad b = 2; \quad f(n) = n \\
& n^{\log_2 1} = 1 \\
& f(n) = n = \Omega(n^{0+1}), \qquad \epsilon = 1 \\
& af(\frac{n}{b}) = \frac{n}{2} \leq cn, \qquad c = \frac{1}{2} < 1 \\
\Rightarrow & f(n) = \Theta(n)
\end{aligned}
$$

f. $3T(\frac{n}{2}) + n$

$$
\begin{aligned}
& a = 3; \quad b = 2; \quad f(n) = n \\
& n^{\log_2 3} = n^{1.58} \\
& f(n) = n = O(n^{1.58-0.58}), \qquad \epsilon = 0.58 \\
\Rightarrow & f(n) = \Theta(n^{\log_2 3})
\end{aligned}
$$

g. $2T(\frac{n}{2}) + n$

$$
\begin{aligned}
& a = 2; \quad b = 2; \quad f(n) = n \\
& n^{\log_2 2} = n \\
\Rightarrow & f(n) = \Theta(n \log n)
\end{aligned}
$$

1.2

Tìm độ phức tạp của từng phương trình và sắp xếp theo thứ tự tăng dần:

a. $T_1(n) = 4T(\frac{n}{2} + 1)$

$$\begin{aligned} & a = 4; \quad b = 2; \quad f(n) = 1 \\ & n^{\log_b a} = n^{\log_2 4} = n^2 \\ & f(n) = O(n^{2-2}), \qquad \epsilon = 2 \\ \Rightarrow & f(n) = \Theta(n^2) \end{aligned}$$

b. $T_2(n) = 4T(\frac{n}{2} + \sqrt{n})$

$$\begin{aligned} & a = 4; \quad b = 2; \quad f(n) = n^{\frac{1}{2}} \\ & n^{\log_b a} = n^{\log_2 4} = n^2 \\ & f(n) = O(n^{2-3/2}), \qquad \epsilon = \frac{3}{2} \\ \Rightarrow & f(n) = \Theta(n^2) \end{aligned}$$

c. $T_3(n) = 4T(\frac{n}{2} + n)$

$$\begin{aligned} & a = 4; \quad b = 2; \quad f(n) = n \\ & n^{\log_b a} = n^{\log_2 4} = n^2 \\ & f(n) = O(n^{2-1}), \qquad \epsilon = 1 \\ \Rightarrow & f(n) = \Theta(n^2) \end{aligned}$$

d. $T_4(n) = 4T(\frac{n}{2} + n^2)$

$$\begin{aligned} & a = 4; \quad b = 2; \quad f(n) = n^2 \\ & n^{\log_b a} = n^{\log_2 4} = n^2 \\ \Rightarrow & f(n) = \Theta(n^2 \log n) \end{aligned}$$

e. $T_5(n) = 4T(\frac{n}{2} + n^3)$

$$
\begin{aligned}
& a = 4; \quad b = 2; \quad f(n) = n^{\frac{1}{2}} \\
& n^{\log_b a} = n^3 \\
& f(n) = \Omega(n^{2+1}), \qquad \epsilon = 1 \\
& af(\frac{n}{b}) = \frac{4n^3}{8} = \frac{n^3}{2} \leq cn^3, \qquad c = \frac{1}{2} \\
\Rightarrow & f(n) = \Theta(n^3)
\end{aligned}
$$

$\Rightarrow T_1 = T_2 = T_3 < T_4 < T_5$

2 Chia để trị

Chia để trị (Divide and Conquer) là một kỹ thuật thiết kế thuật toán, trong đó ta giải quyết một bài toán đệ quy bằng các áp dụng 3 bước:

1. **Chia (Divide)**: Chia bài toán thành các bài toán con nhỏ hơn cùng loại.
2. **Trị (Conquer)**: Giải quyết các bài toán con đệ quy. Nếu bài toán con đủ nhỏ, giải quyết trực tiếp.
3. **Hợp nhất (Combine)**: Kết hợp các lời giải của các bài toán con để tạo thành lời giải cho bài toán gốc.

2.1

Cho một số nguyên X và một mảng số nguyên gồm m phần tử. Hãy thiết kế một thuật toán chia để trị để đếm số lần xuất hiện của X trong mảng.
Mô tả các bước và phân tích độ phức tạp thời gian của thuật toán

Bài làm

```
int cnt_search(int *a, int x, int l, int r) {
    if (l >= r) {
        if (a[l] == x) return 1;
        else return 0;
    }
    int m = (l + r) / 2;
    if (a[m] == x) return 1;
    return cnt_search(a, x, l, m) + cnt_search(a, x, m + 1, r);
}
```

Thời gian chạy:

$$
\begin{aligned}
&T(n) = 2T(\frac{n}{2}) + 1 \\
&a = 2, \quad b = 2; \quad f(n) = 1 \\
&f(n) = 1 = O(n^{1-1}), \qquad \epsilon = 1 \\
\Rightarrow &T(n) = \Theta(n)
\end{aligned}
$$

2.2

Cho một số nguyên S và một mảng số nguyên gồm m phần tử. Tìm hai phần tử trong mảng có tổng bằng S. Hãy thiết kế một thuật toán chia để trị để thực hiện bài toán này (giả sử mảng đã được sắp xếp).
Mô tả các bước và phân tích độ phức tạp thời gian của thuật toán.

Bài làm

```
pair<int, int> find_pair(int *a, int s, int l, int r) {
    if (l >= r) return {-1, -1};
    int m = (l + r) / 2;

    auto result = find_pair(a, s, l, m);
    if (result.first != -1) return result;

    result = find_pair(a, s, m + 1, r);
    if (result.first != -1)  return result;

    int i = m;
    int j = m + 1;
    while (i >= l and j <= r) {
        int sum = a[i] + a[j];
        if (sum == s)
            return {i, j};
        else if (sum > s)
            i--;
        else
            j++;
    }
    return {-1, -1};
}
```

Thời gian chạy:

$$
\begin{aligned}
&T(n) = 2T(\frac{n}{2}) + n \\
&a = 2, \quad b = 2; \quad f(n) = n \\
&f(n) = n = O(n) \\
\Rightarrow &T(n) = \Theta(n \log n)
\end{aligned}
$$

2.3

Cho n điểm trên mặt phẳng có tọa độ (x1, y1),(x2, y2), ...,(xn, yn). Hãy thiết kế một thuật toán chia để trị để tìm cặp điểm gần nhau nhất (có khoảng cách Euclidean nhỏ nhất). Mô tả các bước và phân tích độ phức tạp thời gian của thuật toán.

2.4

Giả sử bạn có một mảng gồm n số nguyên dương. Bạn muốn tìm số lớn thứ hai trong mảng. Bạn có thể làm điều này bằng cách sắp xếp mảng và chọn phần tử ở vị trí thứ hai từ cuối. Tuy nhiên, bạn có thể làm tốt hơn thế. Hãy thiết kế một thuật toán sử dụng ít hơn 2n phép so sánh để tìm số lớn thứ hai trong mảng. Chứng minh tính đúng của thuật toán và phân tích độ phức tạp thời gian của nó.

2.5

Bài toán nhân ma trận Cho hai ma trận vuông A và B có kích thước $n \times n$. Hãy thiết kế thuật toán để tính tích ma trận $C = A \times B$.

a. Viết pseudocode cho thuật toán dựa trên định nghĩa của phép nhân ma trận. Xác định độ phức tạp thời gian của thuật toán.

b. Đưa ra ý tưởng của thuật toán chia để trị để tính tích ma trận. Xác định độ phức tạp thời gian của thuật toán.

c. Tìm hiểu về phương pháp Strassen để tính tích ma trận. Đưa ra ý tưởng chính của phương pháp này để giảm số phép nhân ma trận.

d. Viết pseudocode cho phương pháp Strassen và xác định độ phức tạp thời gian của nó.

e. Cài đặt ba phương pháp nêu trên và thử nghiệm với các ma trận có kích thước khác nhau. So sánh thời gian chạy của ba thuật toán.

2.6 Bài toán Maximum Subarray

Xét bài toán sau:

Một nhà đầu tư muốn mua và bán cổ phiếu của một công ty trong khoảng thời gian nhất định để tối đa hóa lợi nhuận. Mỗi ngày, nhà đầu tư được biết giá cổ phiếu và chỉ được mua một cổ phiếu trong một ngày. Nhà đầu tư chỉ được bán cổ phiếu đã mua sau một ngày với mục tiêu tối đa hóa lợi nhuận (giá bán - giá mua).

Dựa vào dữ liệu giá cổ phiếu trong 10 ngày liên tiếp được minh họa ở hình dưới, hãy xác định: **Ngày mua và ngày bán** để đạt được lợi nhuận tối đa. **Lợi nhuận tối đa** có thể đạt được.

a. Viết pseudocode cho thuật toán vét cạn (brute-force) để giải quyết bài toán này theo thời gian $O(n^2)$.

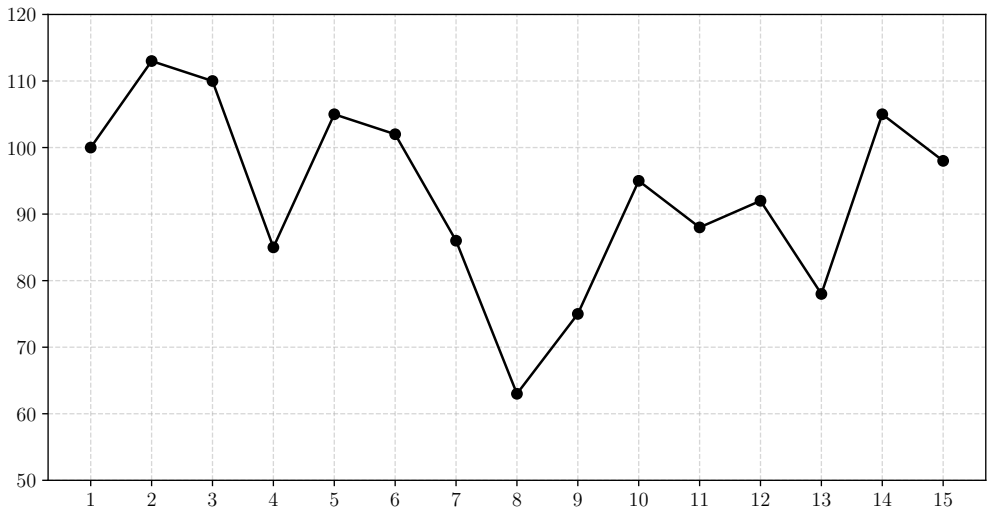
b. Cài đặt thuật toán vét cạn (brute-force) để giải quyết bài toán.

c. Giải thích ý tưởng của thuật toán chia để trị (divide and conquer) để giải quyết bài toán. Xác định độ phức tạp thời gian của thuật toán.

d. Cài đặt thuật toán chia để trị để giải quyết bài toán.

e. Xác định kích thước n_0 để xuất hiện điểm giao (crossover point) mà tại đó thuật toán chia để trị bắt đầu chạy nhanh hơn thuật toán vét cạn.

f. Thay đổi trường hợp cơ sở của thuật toán chia để trị để sử dụng thuật toán vét cạn khi kích thước mảng nhỏ hơn hoặc bằng n_0. Điểm giao mới là bao nhiêu?



Ngày	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Giá	100	113	110	85	105	102	86	63	75	95	88	92	78	105	98