TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

DIVIDE, DECREASE, TRANSFORM AND CONQUER

Môn học: Phân tích và thiết kế thuật toán

Nhóm 9

Sinh viên thực hiện:

Họ và tên	MSSV
Bùi Ngọc Thiên Thanh	23521436
Nguyễn Thái Sơn	23521356

Thành phố Hồ Chí Minh, 2024

Mục lục

1 Bài 1

Hãy đưa ra 3 bài toán có ứng dụng 3 kỹ thuật vừa được học: Divide and conquer, Decrease and conquer, Transform and conquer. Giải thích cụ thể áp dụng như thế nào. (Có thể các bài toán lập trình hoặc các ứng dụng thực tế)

1.1 Divide and Conquer

a. Lý thuyết

Kỹ thuật **Divide and Conquer** chia một bài toán lớn thành nhiều bài toán con độc lập, giải quyết từng bài toán con và sau đó kết hợp kết quả để đưa ra lời giải cho bài toán ban đầu. Đây là phương pháp phổ biến trong lập trình đệ quy và tối ưu hoá thời gian xử lý của nhiều thuật toán phức tạp.

b. Ví dụ ứng dụng

Bài toán: Cho một mảng các số nguyên, tìm phần tử lớn nhất trong mảng.
Áp dụng kỹ thuật:

- **Chia nhỏ:** Chia mảng thành hai nửa.
- **Giải quyết đệ quy:** Tìm phần tử lớn nhất trong từng nửa mảng.
- **Kết hợp:** So sánh giá trị lớn nhất của hai nửa để tìm phần tử lớn nhất trong toàn bộ mảng.

Giải thích: Kỹ thuật Divide and Conquer giúp giảm kích thước của bài toán bằng cách chia mảng ban đầu thành các mảng con nhỏ hơn và giải quyết từng mảng con. Sau đó, ta chỉ cần so sánh kết quả của các mảng con để đưa ra kết quả cuối cùng.

1.2 Decrease and Conquer

a. Lý thuyết

Kỹ thuật **Decrease and Conquer** giải quyết bài toán bằng cách giảm kích thước bài toán trong mỗi bước và giải quyết bài toán con đã giảm kích thước. Thường được thực hiện thông qua đệ quy hoặc vòng lặp, kỹ thuật này đơn giản hóa bài toán bằng cách loại bỏ một phần nhỏ và tái áp dụng giải pháp cho phần còn lại.

b. Ví dụ ứng dụng

Bài toán: Tìm số Fibonacci thứ n, trong đó dãy Fibonacci được định nghĩa như sau:

$$F(0) = 0, \quad F(1) = 1$$

và $F(n) = F(n-1) + F(n-2)$ với $n > 1$.

Áp dụng kỹ thuật:

- **Giảm kích thước:** Đệ quy tính Fibonacci của n bằng cách sử dụng kết quả của $n-1$ và $n-2$.
- **Giải quyết bài toán con:** Lặp lại việc tính toán cho đến khi đạt đến các trường hợp cơ bản $F(0)$ và $F(1)$.
- **Kết hợp:** Cộng kết quả của $F(n-1)$ và $F(n-2)$ để có $F(n)$.

Giải thích: Decrease and Conquer giảm dần giá trị của n trong mỗi lần gọi đệ quy cho đến khi đạt tới các trường hợp cơ bản. Điều này giúp bài toán dần thu nhỏ và đơn giản hơn, đạt được kết quả nhanh hơn.

1.3 Transform and Conquer

a. Lý thuyết

Kỹ thuật **Transform and Conquer** chuyển đổi bài toán hoặc cấu trúc dữ liệu thành một dạng khác dễ xử lý hơn. Sau khi chuyển đổi, bài toán sẽ được giải quyết dựa trên cấu trúc mới này và có thể đạt được lời giải hiệu quả hơn.

b. Ví dụ ứng dụng

Bài toán: Sắp xếp một mảng các số nguyên theo thứ tự tăng dần bằng Heap Sort.

Áp dụng kỹ thuật:

- **Biến đổi cấu trúc dữ liệu:** Chuyển mảng ban đầu thành một cây heap nhị phân (cây hoàn chỉnh có tính chất heap).
- **Giải quyết bài toán:** Áp dụng thuật toán Heap Sort, lần lượt lấy phần tử lớn nhất (hoặc nhỏ nhất) từ cây heap và đặt vào vị trí cuối mảng, sau đó điều chỉnh lại cấu trúc cây để duy trì tính chất heap.
- **Kết hợp:** Lặp lại quá trình cho đến khi tất cả phần tử đã được sắp xếp.

Giải thích: Transform and Conquer sử dụng sự chuyển đổi cấu trúc dữ liệu - từ mảng thành cây heap - để giải quyết bài toán sắp xếp một cách hiệu quả. Bằng cách duy trì cấu trúc heap, Heap Sort giúp tối ưu hóa thời gian xử lý và đạt được hiệu quả $O(n \log n)$.

2 Bài 2

Cho hai số nguyên x và n với điều kiện $x \leq 10^{18}$ và $n \leq 10^{18}$.
Tính tổng:

$$S = x^0 + x^1 + x^2 + x^3 + \ldots + x^n$$

Ví dụ: Với $x = 5$ và $n = 5$, ta có:

$$S = 5^0 + 5^1 + 5^2 + 5^3 + 5^4 + 5^5 = 3906$$

Yêu cầu

- Suy nghĩ bài toán trên có thể có bao nhiêu cách giải (gợi ý có ít nhất 2 cách giải). Với mỗi cách, hãy chỉ ra các kỹ thuật đã học có thể ứng dụng để giải quyết bài toán.
- Viết mã giả (pseudocode) cho các thuật toán mà bạn đã nghĩ ra.

2.1 Cách 1: Tính tổng theo từng lũy thừa (Decrease and Conquer)

Trong cách này, ta tính từng lũy thừa x^i và cộng vào tổng S bằng cách lặp từ $i = 0$ đến $i = n$. Tuy nhiên, với n lớn, phương pháp này có thể tốn nhiều thời gian.

Mã giả:

```
Hàm TinhTongTungLuyThua(x, n):
    S = 0
    x_luythua = 1
    Cho i từ 0 đến n:
        S = S + x_luythua
        x_luythua = x_luythua * x
    Trả về S
```

Ưu và nhược điểm: - **Ưu điểm:** Phương pháp này trực quan và dễ hiểu. - **Nhược điểm:** Độ phức tạp là $O(n)$, nên không phù hợp khi n rất lớn.

2.2 Cách 2: Sử dụng công thức tổng của cấp số nhân (Transform and Conquer)

Dãy $S = x^0 + x^1 + x^2 + \ldots + x^n$ là một cấp số nhân với công bội là x, do đó có thể sử dụng công thức tổng của cấp số nhân:

$$S = \frac{x^{n+1} - 1}{x - 1} \quad (\text{khi } x \neq 1)$$

Nếu $x = 1$, tổng sẽ là $S = n + 1$ (do mọi phần tử trong dãy đều bằng 1).

Ưu điểm: Phương pháp này tính tổng S trong thời gian $O(\log n)$ bằng cách sử dụng lũy thừa nhanh (exponentiation by squaring).

Mã giả:

```
Hàm TinhTong(x, n):
    Nếu x == 1:
```

```
        Trả về n + 1
    Khác:
        Tính x_luythua = x^(n + 1) bằng phương pháp lũy thừa nhanh
        S = (x_luythua - 1) / (x - 1)
        Trả về S
```