Giải bài tập nhóm 8
Nhóm 2: Nguyễn Thiện Nhân, Trần Vinh Khánh

Bài 1

Đề bài: 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 cách áp dụng các kỹ thuật này. (Có thể chọn các bài toán lập trình hoặc các ứng dụng thực tế).

Giải

1. **Divide and Conquer: Tìm kiếm nhị phân trong mảng đã sắp xếp**

- **Mô tả bài toán:** Giả sử ta có một mảng số nguyên đã sắp xếp theo thứ tự tăng dần và cần tìm một giá trị nhất định trong mảng.
- **Cách áp dụng kỹ thuật:** Phương pháp Divide and Conquer chia bài toán thành hai phần nhỏ bằng cách so sánh giá trị cần tìm với phần tử giữa mảng. Nếu giá trị cần tìm nhỏ hơn phần tử giữa, ta tiếp tục tìm trong nửa bên trái; nếu không, ta tìm trong nửa bên phải. Quá trình này lặp lại đến khi tìm thấy giá trị hoặc không còn phần tử để tìm.
- **Lợi ích:** Độ phức tạp của thuật toán giảm từ $O(n)$ xuống $O(\log n)$, nhanh hơn nhiều so với việc kiểm tra từng phần tử trong mảng.

2. **Decrease and Conquer: Tính giai thừa của một số nguyên dương** n

- **Mô tả bài toán:** Tính giai thừa của một số nguyên dương n ($n!$).
- **Cách áp dụng kỹ thuật:** Với mỗi lần tính, ta giảm bài toán về một bài toán con nhỏ hơn, cụ thể là tính $n! = n \cdot (n-1)!$. Quá trình này tiếp tục đến khi đạt đến 1!, là trường hợp cơ sở.
- **Lợi ích:** Kỹ thuật Decrease and Conquer giúp giải bài toán một cách dễ dàng bằng cách giảm kích thước bài toán một cách dần dần.

3. **Transform and Conquer: Sắp xếp một mảng sử dụng thuật toán QuickSort**

- **Mô tả bài toán:** Cho một mảng các số nguyên, yêu cầu sắp xếp mảng theo thứ tự tăng dần.
- **Cách áp dụng kỹ thuật:** QuickSort chọn một phần tử làm pivot và phân chia mảng thành hai phần sao cho các phần tử nhỏ hơn pivot ở bên trái và lớn hơn pivot ở bên phải. Sau đó tiếp tục sắp xếp hai phần nhỏ hơn này. Bài toán được đơn giản hóa thông qua việc biến đổi cấu trúc ban đầu của mảng.
- **Lợi ích:** QuickSort là một thuật toán sắp xếp hiệu quả, trung bình có độ phức tạp $O(n \log n)$.

Bài 2

Đề bài: Cho 2 số nguyên x và n ($x \leq 10^{18}, n \leq 10^{18}$). Tính tổng:

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

Ví dụ: với $x = 5$ và $n = 5$, thì $S = 3906$.

Yêu cầu

- Suy nghĩ bài toán trên có mấy cách giải (Gợi ý có ít nhất 2 cách giải) và đối với mỗi cách hãy cho biết ứng dụng những kĩ thuật gì đã học.
- Viết mã giả cho các thuật toán các bạn đã nghĩ ra ở trên.

Giải

Cách 1: Sử dụng phép lũy thừa trực tiếp (Brute Force)

- **Ý tưởng:** Tính từng lũy thừa của x từ x^0 đến x^n và cộng dồn kết quả.
- **Kỹ thuật sử dụng:** Brute Force - giải thuật đơn giản nhưng không tối ưu cho các giá trị n lớn vì yêu cầu nhiều phép tính.
- **Độ phức tạp:** $O(n)$ vì ta phải tính x^i cho mỗi i từ 0 đến n.

Mã giả:

```
function calculateSum(x, n):
    S = 0
    for i = 0 to n:
        S = S + x^i
    return S
```

Cách 2: Sử dụng công thức cấp số nhân (Efficient Calculation)

- **Ý tưởng:** Dựa vào tính chất của tổng cấp số nhân. Ta có thể sử dụng công thức tổng cấp số nhân cho S:

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

  Nếu $x = 1$, S sẽ là $n + 1$ vì tất cả các lũy thừa đều bằng 1.

- **Kỹ thuật sử dụng:** Transform and Conquer - bằng cách biến đổi bài toán từ phép cộng lũy thừa thành một công thức tổng quát.

- **Độ phức tạp:** $O(\log n)$ nếu sử dụng phương pháp lũy thừa nhanh (fast exponentiation) để tính x^{n+1}.

Mã giả:

```
function calculateSum(x, n):
    if x == 1:
        return n + 1
    else:
        return (x^(n+1) - 1) / (x - 1)
```