ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÀI TẬP NHÓM 8
BỞI NHÓM 12

Giáo viên hướng dẫn: *Nguyễn Thanh Sơn*

Nhóm 12:

- ***Hoàng Minh Thái* - 23521414**
- ***Nguyễn Trọng Tất Thành* - 23521455**

Mục lục

GIẢI BÀI TẬP BỞI NHÓM 12

Bài 1: Các Ứng Dụng Của Các Kỹ Thuật

1.1 Divide and Conquer

Kỹ thuật **Divide and Conquer** (Chia và trị) là một phương pháp giải quyết bài toán bằng cách chia nhỏ bài toán lớn thành những bài toán nhỏ hơn, giải quyết chúng độc lập và sau đó kết hợp kết quả của các bài toán con.

Ví dụ ứng dụng:

- **Thuật toán tìm kiếm nhị phân:** Giả sử bạn có một mảng đã được sắp xếp và bạn muốn tìm một giá trị trong mảng. Thuật toán tìm kiếm nhị phân chia mảng thành hai phần bằng nhau và kiểm tra phần giữa để xác định phần nào chứa giá trị cần tìm. Sau đó, tìm kiếm tiếp tục trong nửa mảng còn lại. Đây là ví dụ điển hình của kỹ thuật `Divide and Conquer`.

- **Sắp xếp Merge Sort:** Thuật toán Merge Sort chia mảng thành các phần nhỏ hơn, sắp xếp các phần đó và sau đó kết hợp chúng lại để tạo ra mảng đã sắp xếp.

1.2 Decrease and Conquer

Kỹ thuật **Decrease and Conquer** (Giảm dần và trị) liên quan đến việc giảm dần kích thước của bài toán trong mỗi bước, tương tự như việc loại bỏ một phần của bài toán trong mỗi bước.

Ví dụ ứng dụng:

- **Tìm kiếm tuyến tính:** Tìm kiếm một phần tử trong mảng bằng cách duyệt qua từng phần tử một. Mỗi lần kiểm tra một phần tử, kích thước bài toán giảm đi một đơn vị.

- **Thuật toán Euclid tính GCD (Greatest Common Divisor):** Thuật toán Euclid tính ước chung lớn nhất của hai số bằng cách chia số lớn hơn cho số nhỏ hơn và thay thế số lớn bằng số dư, tiếp tục cho đến khi dư bằng 0. Kích thước bài toán giảm dần theo mỗi lần chia.

1.3 Transform and Conquer

Kỹ thuật **Transform and Conquer** (Biến đổi và trị) liên quan đến việc biến đổi bài toán thành một dạng khác, sau đó giải quyết bài toán theo cách mới và chuyển đổi lại kết quả vào dạng ban đầu.

Ví dụ ứng dụng:

- **Sắp xếp nhanh Quick Sort:** Thuật toán Quick Sort là một dạng của Transform and Conquer, nơi mảng được phân tách lại dựa trên một chỉ số "pivot". Sau khi phân tách, các phần tử được sắp xếp lại.
- **Thuật toán chuyển đổi giữa hệ cơ số:** Nếu bạn muốn chuyển đổi một số từ hệ thập phân sang hệ nhị phân, bạn có thể sử dụng kỹ thuật biến đổi bằng cách chia số đó cho 2 và lấy phần dư, rồi tiếp tục lặp lại cho đến khi kết quả là 0.

Bài 2: Giải Bài Toán Với Các Kỹ Thuật Đã Học

2.1 Đề Bài

Cho 2 số nguyên x và n ($x \leq 10^{18}$, $n \leq 10^{18}$).
Tính tổng S theo công thức:

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

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

2.2 Các Phương Pháp Giải Bài Toán

Bài toán này có thể giải theo ít nhất hai cách:

- **Cách 1: Sử dụng công thức tổng của cấp số nhân:**

$$S = \frac{x^{n+1} - 1}{x - 1}$$

  Phương pháp này áp dụng kỹ thuật **Transform and Conquer**, biến đổi bài toán thành một công thức tổng quát của cấp số nhân, giúp tính toán nhanh hơn.

- **Cách 2: Sử dụng phương pháp phân chia và trị (Divide and Conquer):** Phương pháp này chia bài toán thành các phần nhỏ, tính giá trị từng phần và kết hợp lại. Cách này phù hợp khi x và n rất lớn, vì nó tránh tính toán trực tiếp các giá trị quá lớn.

2.3 Mã Giả Của Các Thuật Toán

Cách 1: Sử Dụng Công Thức Cấp Số Nhân

```
def sum_geometric_series(x, n):
    if x == 1:
        return n + 1 #  Trng   hp  x = 1,  tng   l  n + 1
    return (x**(n+1) - 1) // (x - 1) # Cng  thc  tng   cp   s   nhn
```

Cách 2: Sử Dụng Phương Pháp Divide and Conquer

```
def sum_powers(x, n):
    if n == 0:
        return 1
    if n % 2 == 0:
        half = sum_powers(x, n // 2)
        return half * (half + 1)
    else:
        return x * sum_powers(x, n - 1)
```

2.4 So Sánh Các Phương Pháp

- Phương pháp 1 nhanh hơn vì nó tính toán trực tiếp từ công thức tổng của cấp số nhân. - Phương pháp 2 có thể chậm hơn, nhưng nó có thể áp dụng cho trường hợp không thể tính toán trực tiếp do giá trị x và n quá lớn.