

**ĐẠI HỌC QUỐC GIA TP.HCM  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ  
THÔNG TIN**



**Khoa Khoa học máy tính**

**Môn học Phân Tích Và Thiết Kế Thuật Toán**

---

**Bài tập Divide, Decrease and Transform and  
Conquer**

---

**Cao Lê Công Thành**

**MSSV: 23521437**

**Đặng Quang Vinh**

**MSSV: 23521786**



## Mục lục

<b>1</b>	<b>Bài tập 1</b>	<b>2</b>
1.1	Kỹ thuật Divide and Conquer - Ứng dụng trong Phân tích Hình ảnh . . . . .	2
1.2	Kỹ thuật Decrease and Conquer - Ứng dụng trong Tìm kiếm Cơ sở Dữ liệu . . . . .	3
1.3	Ứng dụng trong Tối ưu hóa Mạng Lưới . . . . .	4
<b>2</b>	<b>Bài tập 2</b>	<b>5</b>
2.1	Phương pháp Brute Force . . . . .	5
2.2	Phương pháp Lũy thừa nhanh (Fast Exponentiation) . . . . .	5
2.3	Phương pháp sử dụng Công thức cấp số nhân . . . . .	6



## 1 Bài tập 1

### 1.1 Kỹ thuật Divide and Conquer - Ứng dụng trong Phân tích Hình ảnh

Khi xử lý những hình ảnh có độ phân giải cao, như ảnh y tế hay ảnh vệ tinh, sẽ rất khó để máy tính xử lý toàn bộ hình ảnh một cách hiệu quả. Kỹ thuật chia để trị là cách "chia nhỏ" vấn đề ra để xử lý từng phần, giúp máy tính làm việc nhanh hơn. Trong xử lý ảnh, **Quadtree** là một ví dụ của kỹ thuật này.

#### Cách áp dụng:

1. Đầu tiên, hình ảnh lớn được chia thành 4 phần (gọi là *quadrants*). Nếu một vùng nào đó quá phức tạp (có nhiều chi tiết hoặc thay đổi màu sắc), thuật toán sẽ tiếp tục chia nhỏ vùng đó đến khi đạt đến độ chi tiết nhất định.
2. Sau khi đã chia nhỏ, mỗi phần của hình ảnh sẽ được xử lý riêng biệt. Ví dụ, có thể lọc nhiễu trong một vùng nhỏ hoặc nhận diện vật thể trong một vùng khác.
3. Cuối cùng, các phần sau khi đã xử lý sẽ được "gắn" lại để tạo thành hình ảnh hoàn chỉnh.

#### Lợi ích:

- Giúp tối ưu tốc độ xử lý hình ảnh lớn, nhất là khi cần xử lý chi tiết theo từng vùng.
- Rất phù hợp cho các ứng dụng yêu cầu hiệu suất cao như nhận diện khuôn mặt hoặc xử lý video trực tiếp.



## 1.2 Kỹ thuật Decrease and Conquer - Ứng dụng trong Tìm kiếm Cơ sở Dữ liệu

Trong các hệ thống lớn, ví dụ như một trang web thương mại điện tử, việc tìm kiếm dữ liệu cần phải nhanh và chính xác. Một cách làm hiệu quả là sử dụng **Binary Search** - một ứng dụng phổ biến của kỹ thuật giảm dần và chinh phục.

### Cách áp dụng:

1. Dữ liệu được sắp xếp trước (theo thứ tự từ A-Z hoặc tăng dần). Khi có yêu cầu tìm kiếm, hệ thống sẽ bắt đầu ở giữa tập dữ liệu.
2. Nếu giá trị ở giữa lớn hơn hoặc nhỏ hơn giá trị cần tìm, hệ thống sẽ "bỏ qua" một nửa dữ liệu, chỉ tiếp tục tìm trong nửa còn lại.
3. Quá trình này tiếp tục lặp lại, cứ mỗi lần loại bỏ một nửa cho đến khi tìm được kết quả mong muốn.

### Lợi ích:

- Binary Search có độ phức tạp  $O(\log n)$ , nghĩa là với tập dữ liệu càng lớn thì thời gian tìm kiếm cũng chỉ tăng lên một chút. Điều này giúp tăng hiệu suất và giảm tải cho hệ thống, đặc biệt với các cơ sở dữ liệu lớn.



### 1.3 Ứng dụng trong Tối ưu hóa Mạng Lưới

Kỹ thuật **Transform and Conquer** (Biến đổi và Chinh phục) là một phương pháp tối ưu hóa trong các bài toán phức tạp, đặc biệt hữu ích trong các vấn đề tối ưu mạng lưới. Phương pháp này gồm hai bước chính:

1. **Biến đổi (Transform)**: Chuyển bài toán gốc thành một dạng dễ giải quyết hơn.
2. **Chinh phục (Conquer)**: Giải quyết bài toán đã biến đổi và thu được kết quả tối ưu.

Chúng ta sẽ ứng dụng kỹ thuật này trong tối ưu hóa mạng lưới, ví dụ cụ thể là tối ưu hóa mạng lưới truyền tải điện.

Giả sử chúng ta cần tối ưu hóa mạng lưới điện sao cho chi phí truyền tải điện là thấp nhất, trong khi tất cả các trạm phát điện và trạm tiêu thụ đều được kết nối với nhau. Bài toán này có thể được biểu diễn dưới dạng một đồ thị có trọng số, trong đó các đỉnh là các trạm và các cạnh là chi phí truyền tải điện giữa các trạm.

#### Bước 1: Biến đổi bài toán thành đồ thị

Ta có thể biến đổi bài toán mạng lưới điện thành một đồ thị có trọng số:

- Mỗi trạm phát điện và trạm tiêu thụ điện là một đỉnh trong đồ thị.
- Mỗi đoạn đường truyền tải điện là một cạnh giữa hai đỉnh, với trọng số là chi phí truyền tải điện (có thể tính từ khoảng cách hoặc công suất).

#### Bước 2: Áp dụng thuật toán Cây Bao Trùm Nhỏ Nhất (MST)

Một cách để tối ưu hóa chi phí kết nối các trạm điện là sử dụng **cây bao trùm nhỏ nhất (MST)**. Cây bao trùm nhỏ nhất của một đồ thị là cây con bao trùm tất cả các đỉnh với tổng trọng số nhỏ nhất. Để tìm cây bao trùm nhỏ nhất, ta có thể sử dụng một trong hai thuật toán sau:

- **Thuật toán Kruskal**: Sắp xếp các cạnh theo trọng số tăng dần và thêm vào MST nếu không tạo ra chu trình.
- **Thuật toán Prim**: Bắt đầu từ một đỉnh, chọn các cạnh có trọng số nhỏ nhất để mở rộng cây MST.



### Bước 3: Kết hợp và tối ưu

Sau khi tìm được cây bao trùm nhỏ nhất, ta có thể kết hợp các đoạn đường nối để xây dựng mạng lưới truyền tải điện tối ưu. Mạng lưới này sẽ có tổng chi phí thấp nhất mà vẫn đảm bảo kết nối tất cả các trạm phát điện và trạm tiêu thụ điện.

## 2 Bài tập 2

### 2.1 Phương pháp Brute Force

**Kỹ thuật:** Phương pháp cơ bản, tính toán trực tiếp từng lũy thừa  $x^i$  với  $i \in [0, n]$  và cộng dồn chúng lại để có tổng.

**Mã giả:**

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

### 2.2 Phương pháp Lũy thừa nhanh (Fast Exponentiation)

**Kỹ thuật:** Áp dụng kỹ thuật chia để trị (Divide and Conquer) để tính lũy thừa nhanh hơn, giúp tăng tốc độ tính toán các lũy thừa của  $x$  trong quá trình tính tổng.

**Mã giả:**

```
function fast_power(x, i):  
    if i == 0:  
        return 1  
    y = fast_power(x, i // 2)  
    if i is even:  
        return y * y  
    else:  
        return x * y * y
```



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

## 2.3 Phương pháp sử dụng Công thức cấp số nhân

**Kỹ thuật:** Áp dụng phương pháp biến đổi và trị (Transform and Conquer).  
Với  $x \neq 1$ , tổng  $S = x^0 + x^1 + \dots + x^n$  có thể tính bằng công thức tổng cấp số nhân:

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

Mã giả:

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