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 Phân tích độ phức tạp của thuật toán có đệ quy

Cao Lê Công Thành
MSSV: 23521437
Đặng Quang Vinh
MSSV: 23521786

Mục lục

1 Bài toán 1:

Given an input string of numbers, find all combinations of numbers that can be formed using digits in the same order.

1.1 Phân tích bài toán

Bài toán yêu cầu tìm tất cả các dãy con liên tiếp của chuỗi đầu vào. Với một chuỗi có độ dài n, chúng ta cần xét tất cả các dãy con có thể tạo ra từ các chữ số trong chuỗi, đảm bảo rằng các chữ số trong mỗi dãy con phải giữ đúng thứ tự xuất hiện ban đầu. Số lượng các dãy con có thể là $2^n - 1$, bởi vì mỗi chữ số có thể được chọn hoặc không chọn, nhưng không được chọn một chữ số mà bỏ qua các chữ số trước đó.

1.2 Độ phức tạp

Độ phức tạp của thuật toán là $O(2^n)$, bởi vì chúng ta phải tạo ra và kiểm tra tất cả các dãy con có thể, dẫn đến độ phức tạp theo cấp số nhân.

1.3 Phương pháp giải

Một cách tiếp cận là sử dụng phương pháp đệ quy hoặc quay lui để tạo ra tất cả các dãy con theo thứ tự ban đầu. Khi đệ quy, ta sẽ thêm hoặc bỏ qua từng chữ số vào dãy con, và khi độ dài của dãy con đạt yêu cầu, ta sẽ lưu lại kết quả. Ngoài ra, phương pháp lặp cũng có thể được áp dụng bằng cách xem xét từng vị trí trong chuỗi là "bao gồm" hoặc "không bao gồm" trong dãy con.

2 Bài toán 2:

Given a set of characters and a positive integer k, print all possible strings of length k that can be formed from the given set.

2.1 Phân tích bài toán

Bài toán yêu cầu tạo tất cả các chuỗi có độ dài k từ tập hợp các ký tự đã cho, với điều kiện có thể lặp lại ký tự trong chuỗi. Vì vậy, bài toán này là một dạng của **tổ hợp với phép lặp lại**, trong đó mỗi ký tự có thể xuất hiện nhiều lần trong mỗi chuỗi.

2.2 Độ phức tạp

Với tập hợp có m ký tự, có thể tạo ra m^k chuỗi có độ dài k. Do đó, độ phức tạp thời gian của thuật toán là $O(m^k)$, bởi vì tại mỗi vị trí trong chuỗi, ta có m lựa chọn cho ký tự, và ta phải lặp qua tất cả k vị trí.

2.3 Phương pháp giải

Một cách tiếp cận là sử dụng **đệ quy** để tạo ra tất cả các chuỗi. Mỗi cấp độ đệ quy tương ứng với việc thêm một ký tự từ tập hợp vào chuỗi. Bên cạnh đó, có thể sử dụng **vòng lặp lồng nhau**, nhưng đệ quy thường được ưu tiên vì nó đơn giản và dễ triển khai hơn.

3 Bài toán 3:

Write a program to print all the combinations of factors of given number n

3.1 Phân tích bài toán

Bài toán yêu cầu tìm ra tất cả các tổ hợp của các ước (không bao gồm 1 và n) sao cho tích của các ước này bằng n. Ví dụ, với $n = 12$, các tổ hợp sẽ bao gồm: $(2, 6)$, $(3, 4)$, $(2, 2, 3)$, v.v.

3.2 Độ phức tạp

Số lượng ước của n có thể tăng lên theo hàm $O(2^{\sqrt{n}})$ trong trường hợp xấu nhất, vì số tổ hợp của các ước phụ thuộc vào việc phân tích thừa số nguyên tố của n và sự kết hợp các yếu tố này. Mỗi tổ hợp có thể yêu cầu việc kiểm tra nhiều sự kết hợp khác nhau.

3.3 Phương pháp giải

Chúng ta sẽ sử dụng phương pháp đệ quy kết hợp với quay lui để sinh ra tất cả các tổ hợp. Cụ thể:

- Tìm tất cả các ước của n từ 2 đến $\sqrt{n}$.
- Sử dụng đệ quy để thử nghiệm với các tổ hợp của các ước sao cho tích của chúng bằng n. Để làm được điều này, ta cần theo dõi các ước đã chọn trong quá trình đệ quy và kiểm tra nếu chúng có thể kết hợp lại để cho ra n.

4 Bài toán 4

4.1 Phân tích bài toán

- Bài toán yêu cầu tìm các tổ hợp của các số tự nhiên sao cho tổng các lũy thừa bậc n của chúng bằng một số x. Đây là một bài toán có thể coi là một biến thể của bài toán tập con tổng (subset-sum), trong đó các phần tử trong tập hợp cần phải được nâng lên lũy thừa n thay vì sử dụng trực tiếp các số.
- Mục tiêu là tìm các cách biểu diễn x dưới dạng tổng của các lũy thừa bậc n của các số tự nhiên. Phải chú ý rằng các số này phải là các số tự nhiên khác nhau (không lặp lại).
- Ví dụ: Xét ví dụ với $x = 10$ và $n = 2$:
  - Các số tự nhiên có lũy thừa bậc 2 nhỏ hơn 10 là: $1^2 = 1$, $2^2 = 4$, $3^2 = 9$.
  - Tìm các tổ hợp có thể cộng lại thành 10: ta có tổ hợp $1^2 + 3^2 = 1 + 9 = 10$.
  - Ta cần kiểm tra tất cả các khả năng kết hợp các số này sao cho tổng của chúng bằng 10.

4.2 Độ phức tạp

- Độ phức tạp của bài toán này có thể xem là theo cấp số nhân, do phải kiểm tra tất cả các khả năng kết hợp của các số mà lũy thừa n của chúng có thể cộng lại được để tạo ra x. Độ phức tạp trong trường hợp xấu nhất là $O(2^x)$, nhưng khi sử dụng **memoization**, ta có thể giảm thiểu việc tính toán lại những kết quả đã gặp, từ đó tối ưu hóa quá trình tính toán.

4.3 Phương pháp giải

- Một cách hiệu quả để giải bài toán này là dùng phương pháp đệ quy kết hợp với **memoization**. Đối với mỗi số tự nhiên i bắt đầu từ 1, ta kiểm tra xem việc thêm i^n vào tổng hiện tại có thể cho ra giá trị x hay không. Nếu có thể, ta tiếp tục đệ quy với giá trị $x - i^n$. Memoization sẽ lưu trữ các kết quả đã tính toán, giúp tránh việc tính toán lại những giá trị giống nhau nhiều lần.

5 Bài toán 5

5.1 Phân tích bài toán

- Bài toán Tháp Hà Nội yêu cầu giải quyết việc di chuyển các đĩa từ trụ nguồn sang trụ đích. Ta cần di chuyển n đĩa từ trụ nguồn sang trụ đích, thông qua trụ phụ. Mỗi lần di chuyển sẽ thực hiện theo một chiến lược phân chia bài toán thành các phần nhỏ hơn, với bước đi:
  - Di chuyển $n - 1$ đĩa từ trụ nguồn sang trụ phụ.
  - Di chuyển đĩa lớn nhất từ trụ nguồn sang trụ đích.
  - Di chuyển $n - 1$ đĩa từ trụ phụ sang trụ đích.
- Tuy nhiên, quy trình này có thể được tiếp tục chia nhỏ cho các giá trị nhỏ hơn của n, và bài toán trở thành một bài toán đệ quy.
- Ví dụ: Với 3 đĩa, các bước di chuyển có thể diễn ra như sau:
  - Di chuyển 2 đĩa từ trụ nguồn sang trụ phụ.
  - Di chuyển đĩa lớn nhất từ trụ nguồn sang trụ đích.
  - Di chuyển 2 đĩa từ trụ phụ sang trụ đích.
- Mỗi bước lại được chia nhỏ thêm cho 2 đĩa còn lại, tạo thành các bước con, cho đến khi chỉ còn một đĩa cần di chuyển.

5.2 Độ phức tạp

- Độ phức tạp của bài toán Tháp Hà Nội có thể phân tích dựa trên số lượng các bước di chuyển cần thực hiện. Với n đĩa, bài toán yêu cầu di chuyển $n-1$ đĩa hai lần, và mỗi bước đều có thể được chia nhỏ đệ quy. Do đó, tổng số bước di chuyển là:

$$T(n) = 2T(n-1) + 1$$

  Giải phương trình đệ quy này, ta có:

$$T(n) = 2^n - 1$$

  Vì vậy, độ phức tạp thời gian của bài toán là $O(2^n)$, tức là theo cấp số nhân với số lượng đĩa n.

5.3 Phương pháp giải

- Phương pháp giải quyết bài toán này là sử dụng đệ quy. Cách thức giải bài toán có thể mô tả như sau:

  1. Nếu chỉ có một đĩa, ta chỉ cần di chuyển nó từ trụ nguồn sang trụ đích.
  2. Nếu có nhiều hơn một đĩa, ta chia bài toán thành ba bước:
     - Di chuyển $n-1$ đĩa từ trụ nguồn sang trụ phụ (sử dụng trụ đích).
     - Di chuyển đĩa lớn nhất từ trụ nguồn sang trụ đích.
     - Di chuyển $n-1$ đĩa từ trụ phụ sang trụ đích (sử dụng trụ nguồn).
  3. Các bước này sẽ được thực hiện đệ quy cho mỗi nhóm đĩa con.