**Bài tập duyệt toàn bộ từ dễ đến khó**

Xin chào! Tôi sẽ tạo một số bài tập về duyệt toàn bộ (brute force) với độ khó tăng dần. Duyệt toàn bộ là kỹ thuật giải quyết vấn đề bằng cách thử tất cả các khả năng có thể xảy ra.

**Bài tập cơ bản**

**Bài 1: Tìm tổng hai số**

Cho một mảng số nguyên và một số mục tiêu. Hãy tìm hai số trong mảng có tổng bằng số mục tiêu.

Input: nums = [2, 7, 11, 15], target = 9

Output: [0, 1] (vì nums[0] + nums[1] = 2 + 7 = 9)

**Bài 2: Tìm số lớn nhất và nhỏ nhất**

Cho một mảng số nguyên, hãy tìm số lớn nhất và nhỏ nhất trong mảng.

Input: [3, 9, 1, 6, 4, 2]

Output: Max = 9, Min = 1

**Bài 3: Kiểm tra số nguyên tố**

Viết chương trình kiểm tra một số có phải là số nguyên tố hay không bằng cách thử chia số đó cho tất cả các số từ 2 đến căn bậc hai của nó.

Input: 17

Output: Là số nguyên tố

**Bài tập trung bình**

**Bài 4: Sinh tất cả các hoán vị**

Viết chương trình sinh ra tất cả các hoán vị của một mảng n phần tử.

Input: [1, 2, 3]

Output: [1,2,3], [1,3,2], [2,1,3], [2,3,1], [3,1,2], [3,2,1]

**Bài 5: Bài toán người du lịch (TSP đơn giản)**

Cho n thành phố và khoảng cách giữa mỗi cặp thành phố. Hãy tìm đường đi ngắn nhất đi qua tất cả các thành phố, mỗi thành phố chỉ đi qua đúng một lần và quay trở lại thành phố xuất phát.

Input: Ma trận khoảng cách 4x4:

0 10 15 20

10 0 35 25

15 35 0 30

20 25 30 0

Output: Đường đi tối ưu và tổng khoảng cách

**Bài 6: Tìm tập con có tổng bằng k**

Cho một mảng các số nguyên dương và một số k. Hãy kiểm tra xem có tồn tại một tập con nào của mảng có tổng bằng k hay không.

Input: arr[] = {3, 34, 4, 12, 5, 2}, k = 9

Output: True (vì tập con {4, 5} có tổng là 9)

**Bài tập nâng cao**

**Bài 7: Bài toán tám quân hậu**

Đặt 8 quân hậu lên bàn cờ 8x8 sao cho không có quân hậu nào có thể tấn công quân hậu khác. Hãy in ra tất cả các cách đặt có thể.

**Bài 8: Bài toán cái túi (Knapsack)**

Cho n đồ vật, mỗi đồ vật có trọng lượng và giá trị riêng. Hãy chọn các đồ vật để bỏ vào túi sao cho tổng trọng lượng không vượt quá W và tổng giá trị là lớn nhất.

Input:

Giá trị: [60, 100, 120]

Trọng lượng: [10, 20, 30]

W = 50

Output: 220 (chọn đồ vật thứ 2 và 3)

**Bài 9: Sudoku Solver**

Giải bài toán Sudoku bằng phương pháp duyệt toàn bộ với quay lui (backtracking).

Input: Ma trận Sudoku 9x9 với một số ô đã được điền trước

Output: Ma trận Sudoku đã được giải

**Bài tập chuyên sâu**

**Bài 10: Bài toán tô màu đồ thị**

Cho đồ thị G với n đỉnh. Hãy tô màu các đỉnh của đồ thị sao cho hai đỉnh kề nhau không có cùng màu và số màu sử dụng là ít nhất.

Input: Đồ thị G với ma trận kề

Output: Số màu ít nhất cần sử dụng và cách tô màu

**Bài 11: Bài toán phân chia mảng**

Cho một mảng các số nguyên, hãy chia mảng thành k tập con sao cho tổng các phần tử trong mỗi tập con bằng nhau.

Input: arr[] = {4, 3, 2, 3, 5, 2, 1}, k = 4

Output: Có thể chia thành 4 tập con có tổng bằng 5 mỗi tập

**Bài 12: N-Queens tổng quát**

Mở rộng bài toán tám quân hậu lên bàn cờ kích thước nxn với n quân hậu. Hãy đếm số cách đặt có thể và liệt kê một số cách.

Input: n = 10

Output: Số cách đặt 10 quân hậu trên bàn cờ 10x10

Bạn muốn tôi cung cấp mã giải chi tiết cho bài tập nào trước không?

# Bài tập về đệ quy từ dễ đến khó

Xin chào! Tôi sẽ tạo một danh sách các bài tập về đệ quy từ cơ bản đến nâng cao cho bạn.

## Bài tập cơ bản

### Bài 1: Tính giai thừa

Viết hàm đệ quy để tính giai thừa của một số n (n!).

Input: n = 5

Output: 120 (= 5 × 4 × 3 × 2 × 1)

### Bài 2: Dãy Fibonacci

Viết hàm đệ quy để tính số Fibonacci thứ n. Dãy Fibonacci được định nghĩa như sau: F(0) = 0, F(1) = 1, và F(n) = F(n-1) + F(n-2) với n > 1.

Input: n = 6

Output: 8 (dãy Fibonacci: 0, 1, 1, 2, 3, 5, 8, ...)

### Bài 3: Tổng các số từ 1 đến n

Viết hàm đệ quy để tính tổng các số từ 1 đến n.

Input: n = 5

Output: 15 (= 1 + 2 + 3 + 4 + 5)

### Bài 4: Đếm số chữ số

Viết hàm đệ quy để đếm số chữ số của một số nguyên dương.

Input: n = 12345

Output: 5

## Bài tập trung bình

### Bài 5: Tìm UCLN

Viết hàm đệ quy để tìm ước chung lớn nhất (UCLN) của hai số sử dụng thuật toán Euclid.

Input: a = 48, b = 18

Output: 6

### Bài 6: Đảo ngược chuỗi

Viết hàm đệ quy để đảo ngược một chuỗi.

Input: "hello"

Output: "olleh"

### Bài 7: Tính tổng các chữ số

Viết hàm đệ quy để tính tổng các chữ số của một số nguyên dương.

Input: n = 12345

Output: 15 (= 1 + 2 + 3 + 4 + 5)

### Bài 8: Đệ quy nhị phân

Viết hàm đệ quy để chuyển đổi một số nguyên dương sang dạng nhị phân.

Input: n = 10

Output: "1010"

## Bài tập nâng cao

### Bài 9: Tháp Hà Nội

Giải bài toán Tháp Hà Nội bằng đệ quy. Cho n đĩa xếp theo thứ tự kích thước tăng dần từ trên xuống ở cột nguồn. Hãy di chuyển tất cả các đĩa sang cột đích, sử dụng cột trung gian, sao cho mỗi lần chỉ di chuyển một đĩa và không được đặt đĩa lớn hơn lên trên đĩa nhỏ hơn.

Input: n = 3

Output: Các bước di chuyển

### Bài 10: Duyệt mê cung

Viết hàm đệ quy để tìm đường đi từ điểm bắt đầu đến điểm kết thúc trong một mê cung. Mê cung được biểu diễn bằng ma trận, trong đó 0 là ô có thể đi qua, 1 là tường.

Input: Ma trận mê cung, điểm bắt đầu, điểm kết thúc

Output: Đường đi (nếu có)

### Bài 11: Chia để trị - Quicksort

Cài đặt thuật toán sắp xếp nhanh (Quicksort) sử dụng đệ quy.

Input: [5, 2, 9, 1, 7, 6, 3]

Output: [1, 2, 3, 5, 6, 7, 9]

### Bài 12: Tìm tất cả các tập con

Viết hàm đệ quy để liệt kê tất cả các tập con của một tập hợp cho trước.

Input: [1, 2, 3]

Output: [[], [1], [2], [1, 2], [3], [1, 3], [2, 3], [1, 2, 3]]

## Bài tập chuyên sâu

### Bài 13: Bài toán N-Queens

Viết hàm đệ quy để giải bài toán N-Queens: đặt N quân hậu trên bàn cờ NxN sao cho không có quân hậu nào có thể tấn công quân hậu khác.

Input: n = 4

Output: Tất cả các cách đặt quân hậu trên bàn cờ 4x4

### Bài 14: Chia kẹo

Cho n viên kẹo và k người. Hãy chia các viên kẹo cho k người sao cho mỗi người nhận được ít nhất một viên kẹo. Viết hàm đệ quy để đếm số cách chia có thể.

Input: n = 5, k = 3

Output: 6 (Số cách chia 5 viên kẹo cho 3 người)

### Bài 15: Biểu thức đúng

Viết hàm đệ quy để sinh tất cả các biểu thức ngoặc đúng có n cặp ngoặc.

Input: n = 3

Output: ["((()))", "(()())", "(())()", "()(())", "()()()"]

### Bài 16: Sudoku Solver

Viết chương trình giải Sudoku sử dụng đệ quy với kỹ thuật quay lui (backtracking).

Input: Ma trận Sudoku 9x9 chưa hoàn thành

Output: Ma trận Sudoku đã giải (nếu có)

Bạn muốn tôi giải thích chi tiết về bài tập nào trước không?

# Các bài tập Quay lui (Backtracking) từ dễ đến khó

Chào bạn! Dưới đây là danh sách các bài tập quay lui từ mức độ cơ bản đến nâng cao, giúp bạn rèn luyện kỹ năng sử dụng thuật toán quay lui:

## Bài tập cơ bản

1. **Liệt kê các dãy nhị phân độ dài N**
   * Tạo tất cả các dãy nhị phân có độ dài N (chỉ gồm 0 và 1)
   * Ví dụ: N = 3 → 000, 001, 010, 011, 100, 101, 110, 111
2. **Liệt kê hoán vị**
   * Tạo tất cả các hoán vị của N phần tử (1 đến N)
   * Ví dụ: N = 3 → 123, 132, 213, 231, 312, 321
3. **Bài toán N-Queens cơ bản**
   * Đặt N quân hậu trên bàn cờ N×N sao cho không quân nào tấn công quân khác
   * Tìm một cách đặt hợp lệ

## Bài tập trung bình

1. **Tổ hợp chập K của N phần tử**
   * Liệt kê tất cả các cách chọn K phần tử từ tập N phần tử
   * Ví dụ: N=4, K=2 → {1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}
2. **Chia tập hợp**
   * Chia tập N phần tử thành K tập con không rỗng
   * Ví dụ: N=3, K=2 → {{1},{2,3}}, {{2},{1,3}}, {{3},{1,2}}
3. **Sudoku Solver**
   * Giải một bảng Sudoku 9×9 đã có một số ô được điền trước
4. **Bài toán người du lịch (TSP)**
   * Tìm chu trình Hamilton ngắn nhất trên đồ thị có trọng số

## Bài tập nâng cao

1. **Bài toán tô màu đồ thị**
   * Tô màu các đỉnh của đồ thị sao cho hai đỉnh kề nhau không cùng màu
   * Tìm cách tô với số màu tối thiểu
2. **Bài toán cái túi (Knapsack)**
   * Chọn các đồ vật với giá trị và trọng lượng khác nhau để đặt vào túi có giới hạn trọng lượng
   * Mục tiêu: Tối đa hóa tổng giá trị
3. **Tìm đường đi trong mê cung**
   * Tìm đường đi từ điểm bắt đầu đến điểm kết thúc trong mê cung 2D
   * Có thể yêu cầu tìm đường đi ngắn nhất
4. **Bài toán tổng con (Subset Sum)**
   * Tìm tập con của một tập số sao cho tổng của chúng bằng một giá trị cho trước
5. **Bài toán phân hoạch tập hợp (Partition Problem)**
   * Chia một tập số thành hai tập con có tổng bằng nhau
6. **Bài toán Knight's Tour**
   * Tìm đường đi của quân mã trên bàn cờ vua sao cho nó đi qua mỗi ô đúng một lần
7. **Bài toán N-Queens nâng cao**
   * Đếm tất cả các cách đặt N quân hậu trên bàn cờ N×N
   * Tìm tất cả các cách đặt thay vì chỉ một cách
8. **Bài toán đường Hamilton**
   * Tìm đường đi Hamilton trên đồ thị (đi qua mỗi đỉnh đúng một lần)

**Các bài toán quy hoạch động từ dễ đến khó**

Dưới đây là một số bài toán quy hoạch động (Dynamic Programming) được sắp xếp từ dễ đến khó:

**Mức độ cơ bản**

1. **Dãy Fibonacci**
   * Tính số Fibonacci thứ n (F(n) = F(n-1) + F(n-2) với F(0) = 0, F(1) = 1)
   * Đây là ví dụ đơn giản nhất về quy hoạch động
2. **Bài toán leo cầu thang**
   * Có n bậc thang, mỗi lần có thể bước 1 hoặc 2 bậc
   * Tìm số cách khác nhau để leo lên đỉnh
3. **Tổng con liên tiếp lớn nhất (Kadane's Algorithm)**
   * Cho một mảng số nguyên, tìm dãy con liên tiếp có tổng lớn nhất
4. **Đường đi ngắn nhất trong lưới**
   * Cho lưới kích thước m×n, chỉ được đi xuống hoặc sang phải
   * Tìm đường đi từ góc trên trái đến góc dưới phải

**Mức độ trung bình**

1. **Bài toán cái túi 0-1 (Knapsack Problem)**
   * Cho n đồ vật với trọng lượng và giá trị, túi có giới hạn trọng lượng
   * Chọn các đồ vật để bỏ vào túi sao cho tổng giá trị lớn nhất
2. **Dãy con tăng dài nhất (LIS - Longest Increasing Subsequence)**
   * Cho một dãy số, tìm dãy con không nhất thiết liên tiếp dài nhất mà các phần tử tăng dần
3. **Chuỗi con chung dài nhất (LCS - Longest Common Subsequence)**
   * Cho hai chuỗi, tìm chuỗi con chung dài nhất
4. **Chia mảng thành các phần có tổng bằng nhau**
   * Cho một mảng số nguyên, kiểm tra xem có thể chia thành hai phần có tổng bằng nhau không
5. **Cắt thanh (Rod Cutting)**
   * Cho một thanh có chiều dài n và bảng giá của các đoạn có độ dài khác nhau
   * Tìm cách cắt thanh để thu được lợi nhuận lớn nhất

**Mức độ nâng cao**

1. **Tô màu đồ thị (Graph Coloring)**
   * Tô màu đồ thị sao cho hai đỉnh kề nhau có màu khác nhau
   * Tìm số màu tối thiểu cần dùng
2. **Bài toán người du lịch (Traveling Salesman Problem - TSP)**
   * Cho n thành phố và khoảng cách giữa chúng
   * Tìm chu trình ngắn nhất đi qua tất cả các thành phố một lần
3. **Chia mảng thành k phần có tổng bằng nhau**
   * Cho một mảng số nguyên, chia thành k phần có tổng bằng nhau
4. **Bài toán xếp balo nhiều lần (Unbounded Knapsack)**
   * Giống bài toán cái túi nhưng mỗi đồ vật có thể được chọn nhiều lần
5. **Bài toán đoạn palindrome (Palindromic Substrings)**
   * Chia một chuỗi thành các đoạn sao cho mỗi đoạn là một palindrome
   * Tìm số đoạn ít nhất

**Mức độ chuyên sâu**

1. **Quy hoạch động trên cây (Tree DP)**
   * Các bài toán như đường đi độc lập lớn nhất trên cây
   * Tìm tập đỉnh độc lập lớn nhất trên cây
2. **Quy hoạch động với bit mask**
   * Sử dụng biểu diễn nhị phân để lưu trạng thái
   * Ví dụ: Bài toán TSP với bit mask
3. **Bài toán số ma phương (Magic Square)**
   * Tìm ma trận n×n với các số từ 1 đến n² sao cho tổng mỗi hàng, cột và đường chéo đều bằng nhau
4. **Bài toán tô màu Sudoku**
   * Giải bài toán Sudoku sử dụng quy hoạch động

Bạn muốn tôi giải thích chi tiết về bài toán nào cụ thể không?