

Quy hoạch động (Dynamic Programming)

1. Ý nghĩa tên

Thuật ngữ “quy hoạch động” (Dynamic Programming – DP) do Richard Bellman đặt ra vào thập niên 1950. Từ 'programming' ở đây không mang nghĩa 'lập trình máy tính', mà là 'lập kế hoạch' (planning). “Dynamic” nghĩa là 'động', ám chỉ quá trình tối ưu liên tiếp qua các giai đoạn. Do đó, quy hoạch động nghĩa là **lập kế hoạch tối ưu theo từng bước, dựa trên kết quả của các bước trước đó**.

2. Nguyên lý của quy hoạch động

Quy hoạch động dựa trên nguyên lý tối ưu của Bellman:

“Một lời giải tối ưu của bài toán có thể được cấu thành từ các lời giải tối ưu của các bài toán con.”

Cách tiếp cận:

- Chia bài toán lớn thành nhiều bài toán con nhỏ hơn.
- Giải từng bài toán con một lần và lưu kết quả (memoization hoặc bảng DP).
- Dùng lại kết quả đó để xây dựng lời giải cho bài toán lớn hơn.

Điều kiện áp dụng:

- Bài toán có tính chất **con chồng chéo (overlapping subproblems)**.
- Có **cấu trúc con tối ưu (optimal substructure)**.

3. Hai bài toán ví dụ

Ví dụ 1: Dãy Fibonacci

Bài toán: Tính số Fibonacci thứ n.

Công thức đệ quy: $F(n) = F(n-1) + F(n-2)$, với $F(0)=0$, $F(1)=1$.

Nếu dùng đệ quy thuần túy, có rất nhiều phép tính lặp lại. Bằng quy hoạch động, ta lưu kết quả của các $F(i)$ đã tính để tái sử dụng:

$$DP[i] = DP[i-1] + DP[i-2]$$

Ví dụ 2: Balo 0-1 (0-1 Knapsack)

Bài toán: Cho n món đồ, mỗi món có trọng lượng $w[i]$ và giá trị $v[i]$. Một balo có sức chứa tối đa W. Chọn các món sao cho tổng giá trị lớn nhất mà không vượt quá W.

Công thức quy hoạch động:

$$DP[i][j] = \max(DP[i-1][j], DP[i-1][j-w[i]] + v[i]) \text{ nếu } j \geq w[i];$$

$$\text{ngược lại } DP[i][j] = DP[i-1][j].$$

Ý nghĩa: $DP[i][j]$ là giá trị lớn nhất có thể đạt được khi xem xét i món đầu tiên với sức chứa j .

4. Kết luận

Quy hoạch động là một kỹ thuật nền tảng trong khoa học máy tính, giúp giải các bài toán tối ưu hiệu quả hơn so với đệ quy thuần túy. Bản chất của nó là **lưu trữ và tái sử dụng kết quả trung gian**, giảm đáng kể thời gian tính toán.

Một khi hiểu rõ nguyên lý và nhận diện được cấu trúc con tối ưu, DP trở thành công cụ mạnh mẽ trong nhiều lĩnh vực như tối ưu, AI, kinh tế, và sinh học tính toán.