Giải thuật qui hoạch động (Dynamic Programming)

Giải thuật Qui hoạch động (Dynamic Programming) là gì ?

Giải thuật **Qui hoạch động (Dynamic Programming)** giống như giải thuật **chia để trị (Divide and Conquer)** trong việc chia nhỏ bài toán thành các bài toán con nhỏ hơn và sau đó thành các bài toán con nhỏ hơn nữa có thể. Nhưng không giống chia để trị, các bài toán con này không được giải một cách độc lập. Thay vào đó, kết quả của các bài toán con này được lưu lại và được sử dụng cho các bài toán con tương tự hoặc các **bài toán con gối nhau (Overlapping Sub-problems)**.

Chúng ta sử dụng Qui hoạch động (Dynamic Programming) khi chúng ta có các bài toán mà có thể được chia thành các bài toán con tương tự nhau, để mà các kết quả của chúng có thể được tái sử dụng. Thường thì các giải thuật này được sử dụng cho tối ưu hóa. Trước khi giải bài toán con, giải thuật Qui hoạch động sẽ cố gắng kiểm tra kết quả của các bài toán con đã được giải trước đó. Các lời giải của các bài toán con sẽ được kết hợp lại để thu được lời giải tối ưu.

Do đó, chúng ta có thể nói rằng:

- Bài toán ban đầu nên có thể được phân chia thành các bài toán con gối nhau nhỏ hơn.
- Lời giải tối ưu của bài toán có thể thu được bởi sử dụng lời giải tối ưu của các bài toán con.
- Giải thuật Qui hoạch động sử dụng phương pháp lưu trữ (Memoization) – tức là chúng ta lưu trữ lời giải của các bài toán con đã giải, và nếu sau này chúng ta cần giải lại chính bài toán đó thì chúng ta có thể lấy và sử dụng kết quả đã được tính toán.

So sánh

Giải thuật tham lam và giải thuật qui hoạch động

- Giải thuật tham lam (Greedy Algorithms) là giải thuật tìm kiếm, lựa chọn giải pháp tối ưu địa phương ở mỗi bước với hi vọng tìm được giải pháp tối ưu toàn cục.

- Giải thuật Qui hoạch động tối ưu hóa các bài toán con gối nhau.

Giải thuật chia để trị và giải thuật Qui hoạch động:

- Giải thuật chia để trị (Divide and Conquer) là kết hợp lời giải của các bài toán con để tìm ra lời giải của bài toán ban đầu.

- Giải thuật Qui hoạch động sử dụng kết quả của bài toán con và sau đó cố gắng tối ưu bài toán lớn hơn. Giải thuật Qui hoạch động sử dụng phương pháp lưu trữ (**Memoization**) để ghi nhớ kết quả của các bài toán con đã được giải.

Ví dụ giải thuật Qui hoạch động

Dưới đây là một số bài toán có thể được giải bởi sử dụng giải thuật Qui hoạch động:

- Dãy Fibonacci
- Bài toán tháp Hà Nội (Tower of Hanoi)
- Bài toán ba lô
- ...

Giải thuật Qui hoạch động có thể được sử dụng trong cả hai phương pháp **Phân tích (Top-down)** và **Qui nạp (Bottom-up)**. Và tất nhiên là nếu dựa vào vòng đời làm việc của CPU thì việc tham chiếu tới kết quả của lời giải trước đó là ít tốn kém hơn việc giải lại bài toán.