



**HCMC UNIVERSITY OF EDUCATION
FACULTY OF INFORMATION TECHNOLOGY**

QUY HOẠCH ĐỘNG

TS. NGÔ QUỐC VIỆT
2015



Nội dung

1. Giới thiệu
2. Giải quyết một số bài toán bằng quy hoạch động.
 - Minh họa bài toán ba lô 0-1
3. Bài tập
4. DP cho Sequence Alignment
5. Hỏi đáp.



Giới thiệu

- Quy hoạch động (dynamic programming) nhằm giải quyết bài toán tìm:

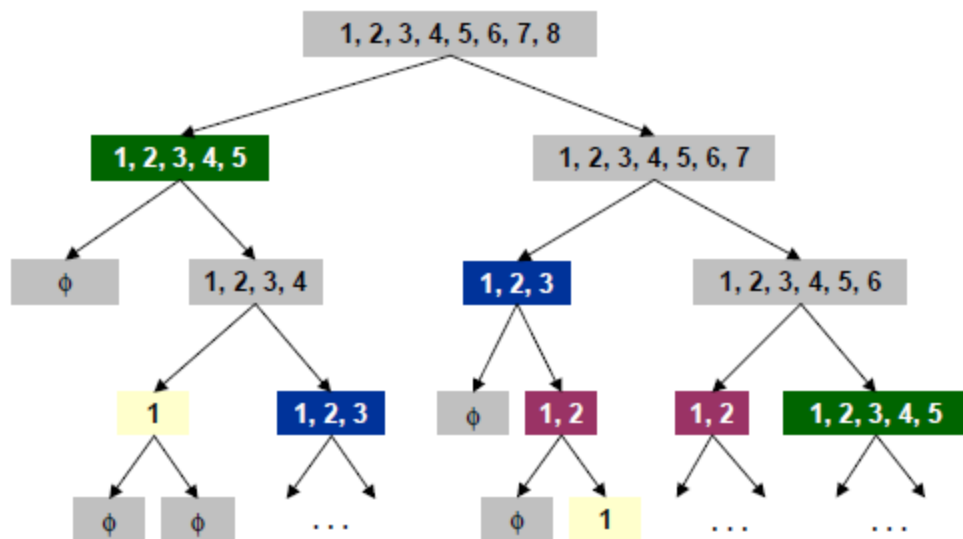
$$\min_{u \in U} g(u)$$

- Trong đó, u là biến (một hay nhiều chiều), $g(u)$ là hàm lượng giá, U là tập điều kiện ràng buộc.
- Là thuật giải dạng **bottom-up**. Nghĩa là các bài toán “con” (không phải được chia từ bài toán lớn theo dạng chia để trị) được giải trước, và tổng hợp để ra kết quả của bài toán lớn.

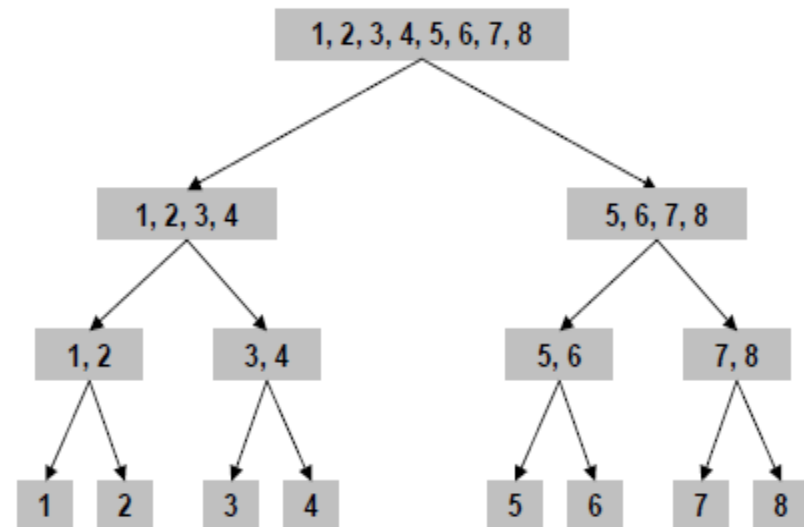


So sánh giữa chia để trị và DP

DP: lời giải con phụ thuộc \rightarrow thuật giải luỹ thừa



Chia để trị: lời giải con độc lập.





Các bước giải quyết

- Xây dựng lời giải cho từng bài toán nhỏ.
- Xây dựng công thức hồi quy nhằm xác định lời giải ở các bước sau.
- Để tránh độ phức tạp lũy thừa \rightarrow lưu trữ các lời giải con vào bảng tra (để không phải giải lại như đệ quy).



Minh hoạ DP với bài toán Knapsack

- DỮ LIỆU: n item
 - w_i = cân nặng của item i
 - v_i = giá trị của item i
 - W = sức chứa của knapsack
- GIẢI PHÁP: tìm tập con S các item sao cho trọng lượng không vượt quá W
- MỤC TIÊU: tìm cực đại giá trị của S



Bài toán Knapsack

- Nghĩa là tìm: tập con S sao cho

$$\text{maximizes } \sum_{i \in S} v_i$$

$$\sum_{i \in S} w_i \leq W.$$

- Được gọi là bài toán 0-1 Knapsack.
- Chú ý: tìm cực đại theo giá trị thay vì theo trọng lượng hay kích thước của balô.



Thuật giải đơn giản

- Sắp xếp các items theo “price per pound”

$$v_i/w_i$$

- Lấy từng item theo thứ tự này bỏ vào balô, nếu vẫn còn bỏ được (chú ý đến điều kiện ràng buộc).
- Sinh viên hãy đánh giá thuật giải này (có tìm được KQ tối ưu không?)



Sử dụng DP cho Knapsack 0-1

- Minh họa kết quả khác biệt giữa thuật giải đơn giản (tham lam) và DP

v_i / w_i

Item	Value	Weight
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

$W = 11$

Greedy = 35: { 5, 2, 1 }

OPT value = 40: { 3, 4 }



Sử dụng DP cho Knapsack 0-1

- Đặt $K[j]$ là giá trị tối ưu của các item chứa trong ba lô kích thước j ($< W$).
- Yêu cầu cuối cùng cần tìm là $K[M]$.
- Để tìm $K[W, n]$, cần xác định các lời giải con bao gồm i ($0 < i < n$) item đầu tiên cho ba lô có kích thước ($0 < j < W$).