

CS112.L21.KHCL – Nguyễn Thế Sơn

Bài tập Dynamic Programming: KnapSack Problem

Nhóm 6:

Trần Tiến Hưng – 19521587

Lê Vinh Quang – 19522093

Bùi Thị Bích Hậu – 19521483

Trạng thái:

- Sử dụng memoization và đệ qui.
- Tạo ra mảng 2D để thể hiện trạng thái  $(n, w)$  cho lần tính đầu tiên.
- Nếu có gặp lại trạng thái  $(n, w)$  đó thì sẽ trả về giá trị thay vì phải tính lại.
- Cho 2 mảng  $val[0..n-1]$  và  $wt[0..n-1]$  lần lượt là giá trị của hàng hóa và khối lượng hàng hóa.
- Cho số nguyên  $W$  thể hiện khối lượng max mà túi có thể chứa.

Mối quan hệ:

- Output mảng con với max val ứng với tổng khối lượng mà bé hơn hoặc bằng  $W$ .
- Không thể cắt bớt khối lượng cũng như giá trị.
- Không thể lấy lại hàng hóa đó.

Base case:

- If  $(n==0)$  and  $(w==0)$ : return 0
- If  $t[n][w] \neq -1$ : return  $t[n][w]$

Code:

```
def knapsack(wt, val, W, n):  
  
    # base conditions  
    if n == 0 or W == 0:  
        return 0  
    if t[n][W] != -1:  
        return t[n][W]  
  
    # choice diagram code  
    if wt[n-1] <= W:  
        t[n][W] = max(  
            val[n-1] + knapsack(  
                wt, val, W-wt[n-1], n-1),  
            knapsack(wt, val, W, n-1))  
    return t[n][W]
```

```
elif wt[n-1] > W:  
    t[n][W] = knapsack(wt, val, W, n-1)  
    return t[n][W]
```

Nguồn: <https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/>

Giải thích:

- Optimal sub-problem: Để có thể tính đc tập con của các hàng hóa, sẽ có 2 TH cho mỗi hàng hóa.
- Case 1: Vật đó được tính và tập con
- Case 2: Vật đó không được tính và tập chính.
- Để kiểm max val kiểm được từ n vật thì giá trị đó sẽ là max của 2 giá trị:
  - o Max val của vật thứ n-1 và W khối lượng
  - o Giá trị của vật thứ n + max val của vật thứ n-1 và W trừ đi khối lượng của vật thứ n.
- Nếu khối lượng của vật thứ n > W thì vật đó sẽ không được cho vào list.

Độ phức tạp:  $O(N*W)$