

# **Giải quyết bài toán Knapsack bằng Dynamic Programing**

## **1. Các trạng thái**

- Trạng thái được xác định bằng 2 tham số là index và weight.
- $MaxV(i, W)$  là tổng giá trị lớn nhất hiện tại khi chọn đến món đồ thứ  $i$  ( $1 \rightarrow i$ ) với trọng lượng tối đa của cái túi hiện tại là  $W$ .
- $MaxV(n, b)$  là tổng giá trị lớn nhất mang đi được với  $n$  món đồ và  $b$  là tổng khối lượng còn lại.

## **2. Base case**

- Bài toán con:  $Max(i, W)$  với  $i = 0$ .

## **3. Mối liên hệ giữa các trạng thái**

- $Max(i, W) = MaxV(i-1, W)$ : trong trường hợp khi ta chọn vật thứ  $i$  tiếp theo mà weight của món vật thứ  $i$  lại lớn hơn capacity còn lại, thì món đồ thứ  $i$  đó sẽ không được chọn, và giá trị lớn nhất của cái túi lúc này sẽ là giá trị lớn nhất khi ta chọn đồ vật  $i-1$ .
- $Max(i, W) = Max(i, W-w[i]) + v[i]$ : trong trường hợp khi ta chọn vật thứ  $i$  tiếp theo mà weight của món vật đó nhỏ hơn so với capacity còn lại, thì món đồ thứ  $i$  đó sẽ được chọn, và giá trị lớn nhất của cái túi lúc này sẽ là giá trị lớn nhất khi ta chọn với đồ vật thứ  $i$  cộng cho giá trị đồ vật thứ  $i$ , và lúc này capacity sẽ bằng  $W-w[i]$ .