CANTHO UNIVERSITY

CHƯƠNG 3

KỸ THUẬT THIẾT KẾ THUẬT TOÁN

KỸ THUẬT QUY HOẠCH ĐỘNG
(Dynamic Programming)

Võ Huỳnh Trâm

Một số kỹ thuật thiết kế thuật toán

- Kỹ thuật **Chia để trị** (Divide and Conquer)
- Kỹ thuật **Tham ăn** */Háu ăn/Tham lam* (Greedy)
- Kỹ thuật **Nhánh cận** (Branch and Bound)
- Kỹ thuật **Quy hoạch động** (Dynamic Programming)
- Kỹ thuật **Quay lui** (Backtracking)
- Kỹ thuật **Cắt tỉa alpha-beta** (Alpha-Beta Pruning) trên Cây trò chơi
- *Kỹ thuật **Tìm kiếm địa phương** (Local Search)*

Thuật toán Quy hoạch động

- **Ý tưởng** : Trong thuật toán đệ quy, một số bài toán con phải giải **nhiều lần**.

→ **Giải pháp**: Tạo **bảng lưu trữ** kết quả các bài toán con để khi cần sẽ sử dụng mà **không cần phải giải lại**.

Thuật toán Quy hoạch động : 2 bước

(1) **Tạo bảng** bằng cách:

- Gán giá trị cho một số ô nào đó.
- Gán trị cho các ô khác nhờ vào giá trị của các ô trước đó.

(2) **Tra bảng** và xác định kết quả bài toán ban đầu.

Quy hoạch động: Ưu và nhược điểm

- **Ưu điểm:**
  - Chương trình **thực hiện nhanh.**
  - Kỹ thuật quy hoạch động có thể vận dụng để giải các bài toán *tối ưu*, **các bài toán có *công thức truy hồi***.
- **Nhược điểm:** Quy hoạch động không hiệu quả khi:
  - Không tìm được *công thức truy hồi*.
  - Số lượng bài toán con cần giải quyết và lưu giữ kết quả là rất lớn.
  - Sự kết hợp lời giải của các bài toán con chưa chắc cho lời giải của bài toán ban đầu.

Bài toán TÍNH SỐ TỔ HỢP

- **<u>Bài toán</u>:** Tính số ***tổ hợp lặp chập k của n*** theo công thức truy hồi:

$$C^k_n = \begin{cases} 1 & \text{neu } k = 0 \text{ hoac } k = n \\ C^{k-1}_{n-1} + C^k_{n-1} & \end{cases}$$

Bài toán tính số tổ hợp
Thuật toán ĐỆ QUY

```
int Comb(int n, int k) {
  if (k==0 || k==n)
      return 1;
  return Comb(n-1, k-1) + Comb(n-1, k);
}
```

Bài toán tính số tổ hợp
Đánh giá Thuật toán Đệ quy

- Gọi **T(n)** là thời gian để tính số tổ hợp chập k của **n.**

→ **T(n-1)** là thời gian tính số tổ hợp chập i của **(n-1)**

- Khi n =1 thì k= 0 hoặc k = 1= n → **Return 1** tốn $\mathbf{C_1}$
- Khi n >1, trường hợp xấu nhất 0 <k < n → Chương trình **gọi đệ quy 2 tổ hợp chập** con **2T(n-1).** Thời gian thực hiện phép **cộng** và trả về kết quả là hằng $\mathbf{C_2}$
- Ta có phương trình đệ quy:

$$\boxed{\begin{array}{l} T(1) = C_1 \\ T(n) = 2T(n-1) + C_2 \end{array}}$$

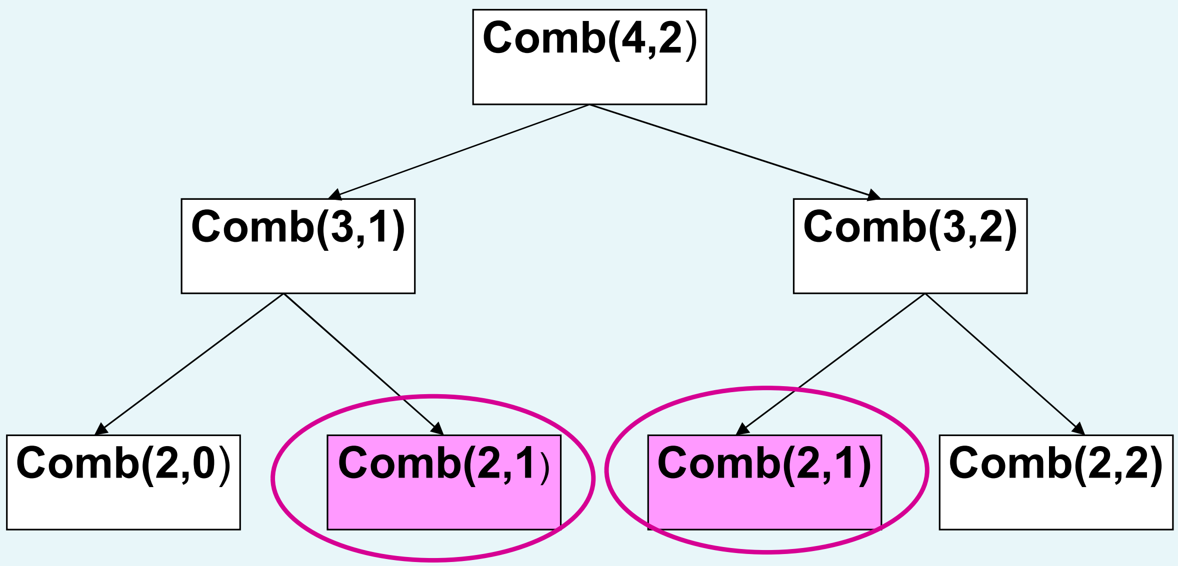
$$\Rightarrow T(n) = O(2^n)$$

Bài toán tính số tổ hợp
Nhận xét Thuật toán Đệ quy

$$C^k_n = \begin{cases} 1 & \text{neu } k = 0 \text{ hoac } k = n \\ & C^{k-1}_{n-1} + C^k_{n-1} \end{cases}$$

Nhận xét:

Ví dụ tính số tổ hợp chập 2 của 4 (***k* = *2*, *n* = *4***)



Bài toán tính số tổ hợp
Thuật toán QUY HOẠCH ĐỘNG

(1) Tạo bảng: Xây dựng một bảng **C** gồm **n+1 dòng** (từ 0 đến n) và **n+1 cột** (từ 0 đến n).

- **C[i,j]** lưu trữ giá trị của **Comb(i,j)** theo quy tắc truy hồi sau: (Quy tắc **tam giác Pascal**):
  - C[0,0] = 1;
  - C[i,0] =1;
  - C[i,i] = 1 với $0 < i \leq n$;
  - **C[i,j] = C[i-1,j-1] + C[i-1,j]**
    với $0 < j < i \leq n$.

(2) Tra bảng: C[n,k] chính là **Comb(n,k)**.

i \ j	0	1	2	3	4
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

Tam giác Pascal

Bài toán tính số tổ hợp
Thuật toán Quy hoạch động

```
int Comb_QHD (int n, int k) {
  int C[n+1,n+1], i, j;
  C[0,0] = 1;
  for (i = 1; i <= n; i++) {
     C[i,0] = 1;
     C[i,i] = 1;
   for (j = 1; j < i; j++)
     C[i,j]=C[i-1,j-1]+C[i-1,j];  }
  return C[n,k];  }
```

i \ j	0	1	2	3	4
0	1				
1					
2					
3					
4					

Bài toán tính số tổ hợp
Thuật toán Quy hoạch động

```
int Comb_QHD (int n, int k) {
  int C[n+1,n+1], i, j;
  C[0,0] = 1;
  for (i = 1; i <= n; i++) {
    C[i,0] = 1;
    C[i,i] = 1;
    for (j = 1; j < i; j++)
      C[i,j]=C[i-1,j-1]+C[i-1,j];  }
  return C[n,k];  }
```

i \ j	0	1	2	3	4
0	1				
1	1	1			
2	1		1		
3	1			1	
4	1				1

Bài toán tính số tổ hợp
Thuật toán Quy hoạch động

```
int Comb_QHD (int n, int k) {
   int C[n+1,n+1], i, j;
   C[0,0] = 1;
   for (i = 1; i <= n; i++) {
      C[i,0] = 1;
      C[i,i] = 1;
      for (j = 1; j < i; j++)
       C[i,j]=C[i-1,j-1]+C[i-1,j];}
   return C[n,k];  }
```

i \ j	0	1	2	3	4
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

CANTHO UNIVERSITY

Bài toán tính số tổ hợp
Thuật toán Quy hoạch động

```
int Comb_QHD (int n, int k) {
  int C[n+1,n+1], i, j;
  C[0,0] = 1;
  for (i = 1; i <= n; i++) {
    C[i,0] = 1;
    C[i,i] = 1;
    for (j = 1; j < i; j++)
     C[i,j]=C[i-1,j-1]+C[i-1,j];  }
  return C[n,k];  }
```

→ C[4,2] = 6

i \ j	0	1	2	3	4
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

Bài toán tính số tổ hợp: Đánh giá thuật toán Quy hoạch động

- Thuật toán quy hoạch động gán trị cho ½ **mảng hai chiều** (n+1) dòng, (n+1) cột với số phần tử là $\mathbf{(n-1)^2/2}$ ô của bảng.
- Mỗi phép gán trị cần 1 đơn vị thời gian (hằng C).
- Gọi T(n) là thời gian thực hiện chương trình:

$$\mathbf{T(n) = (n-1)^2 C/2 = O(n^2)}$$

→ Thuật toán **quy hoạch động** hiệu quả hơn nhiều so với thuật toán **đệ qui** ($\mathbf{n^2 < 2^n}$).

Ví dụ *với n = 10*: Thuật toán đệ quy : $2^n = 1024$

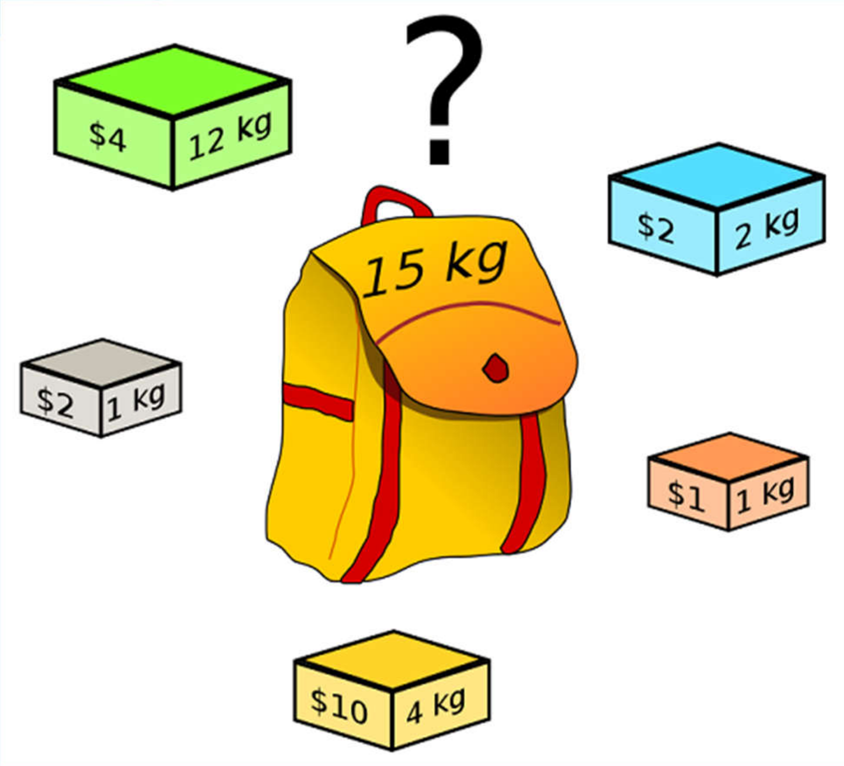
Thuật toán quy hoạch động : $n^2 = 100$

Quy hoạch động: Bài toán Cái ba lô

- **Bài toán**: Cho một cái ba lô có thể chứa trọng lượng **W** với **n** loại đồ vật, mỗi đồ vật **i** có một **trọng lượng g_i** và một **giá trị v_i**. *Tất cả đồ vật đều có số lượng không hạn chế*. Tìm một cách lựa chọn các đồ vật đựng vào ba lô sao cho tổng trọng lượng không vượt quá **W** và tổng giá trị đồ vật là **lớn nhất.**

- **Yêu cầu**: Sử dụng **kỹ thuật quy hoạch động** để giải bài toán cái ba lô với điều kiện các số liệu (*trọng lượng ba lô và trọng lượng các đồ vật)* đều được cho dưới dạng ***số nguyên***.

Bài toán CÁI BA LÔ



Trọng lượng = W CBL 1

Số đồ vật = n

Đồ vật i → TL g_i, GT v_i, SL --

f(X) → Max: CT

Quy hoạch động: Bài toán cái ba lô
- Công thức truy hồi

(1) Tạo bảng: Xây dựng công thức truy hồi

- **k** : đồ vật (***k = 1 .. n)***
- **V**: trọng lượng <u>còn lại </u>của ba lô ***(V = 0 ..W)***

Đặt : **X[k,V]** = **<u>số lượng </u>**đồ vật k được chọn

F[k,V] = **<u>tổng giá trị </u>**k đồ vật đã được chọn

$$\mathbf{F(n,W) = F(X)} = x_1 * v_1 + x_2 * v_2 + \ldots + x_n * v_n \rightarrow \mathbf{Max}$$

Quy hoạch động: Bài toán cái ba lô - Công thức truy hồi

X[k,V] = **số lượng** đồ vật k được chọn

F[k,V] = **tổng giá trị** k đồ vật đã được chọn

- **Trường hợp chỉ có 1 đồ vật (k = 1):**

  Đồ vật 1 : TL g_1, GT v_1

  SL: $X[1,V] = V/g_1$

  TGT: $F[1,V] = X[1,V] * v_1$, $\forall V = 0 .. W$

Quy hoạch động: Bài toán cái ba lô - Công thức truy hồi

X[k,V] = **<u>số lượng</u>** đồ vật k được chọn

F[k,V] = **<u>tổng giá trị</u>** k đồ vật đã được chọn

- **Trường hợp <u>đồ vật</u> k > 1: Đồ vật k : TL g_k, GT v_k**

Đã có **F[k-1,V]**, khi có thêm đồ vật thứ k→ tính **F[k,V]**, ∀V = 0 .. W.

Cách tính: Đặt $y_k = V/g_k$ (số lượng tối đa đồ vật k có thể bỏ vào ba lô)

⇒ Các khả năng chọn đồ vật k là x_k : **0 →** $y_k = V/g_k$. Nếu chọn x_k thì :

- Trọng lượng còn lại của ba lô là : $U = V - x_k * g_k$

- Tổng giá trị k loại đồ vật đã chọn $F[k,V] = F[k-1,U] + x_k * v_k$,

với x_k **đổi từ 0 đến** $y_k = V/g_k$ và chọn x_k sao cho **F[k,V]** **<u>lớn nhất</u>**.

Quy hoạch động: Bài toán cái ba lô - CÔNG THỨC TRUY HỒI

- **Công thức truy hồi** như sau:

$X[1,V] = V/g_1$ và $F[1,V] = X[1,V] * v_1$

$F[k,V] = Max(F[k-1,V-x_k*g_k] + x_k*v_k)$,

với $x_k = 0..V/g_k$

- Khi xác định được **F[k,V]** thì **X[k,V]** là x_k ứng với **F[k,V]** được chọn.

Quy hoạch động: Bài toán cái ba lô - TẠO BẢNG

- Tạo bảng:

- **n dòng** (1 .. n): dòng thứ k ứng với **đồ vật k**

- **W+1 cột** (0 ..W): cột thứ V ứng với **trọng lượng còn lại V**

F[k, V] :TGT

X[k, V]: SL

- Mỗi cột V gồm 2 cột:

Bên trái **F[k,V] : TGT**, bên phải **X[k,V] : SL**

(Khi lập trình, sẽ tổ chức thành 2 bảng F và X)

Quy hoạch động: Bài toán cái ba lô - Ví dụ

Ví dụ : Cho bài toán cái ba lô với trọng lượng **W=9**, và 5 loại đồ vật được cho trong bảng sau: **(n=5, W=9)**

Đồ vật	Trọng lượng (g_i)	Giá trị (v_i)
1	3	4
2	4	5
3	5	6
4	2	3
5	1	1

Bảng F và X với W=9

n = 5 dòng (1 .. 5): ứng với 5 loại đồ vật

W+1 = 9 + 1 = 10 cột (0 ..9): ứng với trọng lượng còn lại V

Đồ vật	g_i	v_i
1	3	4
2	4	5
3	5	6
4	2	3
5	1	1

F[k, V] :TGT

X[k, V]: SL

k \ V	0		1		2		3		4		5		6		7		8		9	
1	0	0	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3
2	0	0	0	0	0	0	4	0	5	1	5	1	8	0	9	1	10	2	12	0
3	0	0	0	0	0	0	4	0	5	0	6	1	8	0	9	0	10	0	12	0
4	0	0	0	0	3	1	4	0	6	2	7	1	9	3	10	2	12	4	13	3
5	0	0	1	1	3	0	4	0	6	0	7	0	9	0	10	0	12	0	13	0

Cách tính bảng F và X

F[k, V] X[k, V]

k \ V	0		1		2		3		4		5		6		7		8		9	
1	0	0	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3
2	0	0	0	0	0	0	4	0	5	1	5	1	8	0	9	1	10	2	12	0
3	0	0	0	0	0	0	4	0	5	0	6	1	8	0	9	0	10	0	12	0
4	0	0	0	0	3	1	4	0	6	2	7	1	9	3	10	2	12	4	13	3
5	0	0	1	1	3	0	4	0	6	0	7	0	9	0	10	0	12	0	13	0

- Điền giá trị cho **dòng 1** theo:

  $X[1,V] = V \text{ DIV } g_1$ và **$F[1,V] = X[1,V] * v_1$.**

- Từ ***dòng 2 đến dòng 5***, sử dụng công thức truy hồi:

  $F[k,V] = Max(F[k-1,V-x_k*g_k] + x_k*v_k)$

  với $x_k : 0 \rightarrow V / g_k$

Bảng F và X: Dòng 1

$X[1,7] = V / g_1 = 7/ 3 = 2$

$F[1,7] = X[1,7] * v_1 = 2 * 4 = 8$

Đồ vật	g_i	v_i
1	3	4
2	4	5
3	5	6
4	2	3
5	1	1

- Điền giá trị cho dòng 1:

  X[1,V] = V DIV g_1 và **F[1,V] = X[1,V] * v_1**.

F[1,7] X[1,7]

k \ V	0		1		2		3		4		5		6		7		8		9	
1	0	0	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3
2	0	0	0	0	0	0	4	0	5	1	5	1	8	0	9	1	10	2	12	0
3	0	0	0	0	0	0	4	0	5	0	6	1	8	0	9	0	10	0	12	0
4	0	0	0	0	3	1	4	0	6	2	7	1	9	3	10	2	12	4	13	3
5	0	0	1	1	3	0	4	0	6	0	7	0	9	0	10	0	12	0	13	0

k \ V	0		1		2		3		4		5		6		7		8		9	
1	0	0	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3
2	0	0	0	0	0	0	4	0	5	1	5	1	8	0	9	1	10	2	12	0
3	0	0	0	0	0	0	4	0	5	0	6	1	8	0	9	0	10	0	12	0
4	0	0	0	0	3	1	4	0	6	2	7	1	9	3	10	2	12	4	13	3
5	0	0	1	1	3	0	4	0	6	0	7	0	9	0	10	0	12	0	13	0

Ví dụ: Tính **F[2,7] và X[2,7]** **?**

(Đồ vật k=2: $g_2 = 4$, $v_2 = 5$, *trọng lượng ba lô còn lại V=7)*

F[k,V] = Max(F[k-1,V-x_k*g_k] + x_k*v_k) với **x_k : 0 → V / g_k.**

Ta có x_2 : 0 → V / g_2 = 7 / 4 = 1, tức x_2 có 2 giá trị **0 và 1.**

Khi đó **F[2,7]** = **Max** (F[2-1, 7-0*4] + 0*5, F[2-1,7-1*4] + 1*5)

= **Max**(**F[1,7], F[1,3]** + 5) = Max(8, 4+5) = Max(8, 9) = **9**.

F[2,7] = 9 ứng với x_k = 1 do đó **X[2,7]** = **1**.

Ví dụ: Tính F[4, 8] và X[4, 8] ?

k \ V	0		1		2		3		4		5		6		7		8		9	
1	0	0	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3
2	0	0	0	0	0	0	4	0	5	1	5	1	8	0	9	1	10	2	12	0
3	0	0	0	0	0	0	4	0	5	0	6	1	8	0	9	0	10	0	12	0
4	0	0	0	0	3	1	4	0	6	2	7	1	9	3	10	2	12	4	13	3
5	0	0	1	1	3	0	4	0	6	0	7	0	9	0	10	0	12	0	13	0

Ví dụ: Tính **F[4, 8] và X[4, 8]** ?

(Đồ vật k=4: $g_4 = 2$, $v_4 = 3$,

trọng lượng ba lô còn lại V=8)

Đồ vật	g_i	v_i
1	3	4
2	4	5
3	5	6
4	2	3
5	1	1

Ví dụ: Tính F[4, 8] và X[4, 8]

Đồ vật $k=4$: $g_4 = 2$, $v_4 = 3$, *trọng lượng ba lô còn lại $V=8$*

$F[k,V] = Max(F[k-1,V-x_k*g_k] + x_k*v_k)$ với $x_k : 0 \rightarrow V / g_k$

Ta có $x_4 : 0 \rightarrow V / g_4 = 8 / 2 = 4$, tức x_4 có 5 giá trị **0, 1, 2, 3, 4.**

$$
\begin{aligned}
F[4, 8] = Max (&F[4 - 1, 8 - \mathbf{0} * 2] + \mathbf{0} * 3, \\
&F[4 - 1, 8 - \mathbf{1} * 2] + \mathbf{1} * 3, \\
&F[4 - 1, 8 - \mathbf{2} * 2] + \mathbf{2} * 3, \\
&F[4 - 1, 8 - \mathbf{3} * 2] + \mathbf{3} * 3, \\
&F[4 - 1, 8 - \mathbf{4} * 2] + \mathbf{4} * 3)
\end{aligned}
$$

$$
\begin{aligned}
= Max (&F[3 ,8] + 0, F[3, 6]+ 3, F[3, 4]+ 6, \\
&F[3, 2]+ 9, F[3, 0]+ 12)
\end{aligned}
$$

CANTHO UNIVERSITY

k \ V	0		1		2		3		4		5		6		7		8		9	
1	0	0	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3
2	0	0	0	0	0	0	4	0	5	1	5	1	8	0	9	1	10	2	12	0
3	0	0	0	0	0	0	4	0	5	0	6	1	8	0	9	0	10	0	12	0
4	0	0	0	0	3	1	4	0	6	2	7	1	9	3	10	2	12	4	13	3
5	0	0	1	1	3	0	4	0	6	0	7	0	9	0	10	0	12	0	13	0

*Đồ vật **k=4***: $g_4 = 2$, $v_4 = 3$, *trọng lượng ba lô còn lại **V=8***

$$F[4,8] = \text{Max}\,(F[3,8] + 0,\ F[3,6] + 3,\ F[3,4] + 6,\ F[3,2] + 9,\ F[3,0] + 12)$$

$$= \text{Max}\,(10,\ 8 + 3,\ 5 + 6,\ 0 + 9,\ 0 + 12)$$

$$= \text{Max}\,(10, 11, 11, 9, 12) = 12$$

F[4,8] = 12 ứng với $x_4 = 4$ do đó **X[4, 8] = 4**.

(2) Tra bảng: Xác định phương án

k \ V	0		1		2		3		4		5		6		7		8		9	
1	0	0	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3
2	0	0	0	0	0	0	4	0	5	1	5	1	8	0	9	1	10	2	12	0
3	0	0	0	0	0	0	4	0	5	0	6	1	8	0	9	0	10	0	12	0
4	0	0	0	0	3	1	4	0	6	2	7	1	9	3	10	2	12	4	13	3
5	0	0	1	1	3	0	4	0	6	0	7	0	9	0	10	0	12	0	13	0

- Khởi đầu, trọng lượng còn lại của ba lô V = W.
- Xét các đồ vật từ n đến 1, với mỗi đồ vật k, ứng với trọng lượng còn lại V của ba lô, nếu X[k,V] > 0 thì **chọn X[k,V] đồ vật loại k**. Tính lại **V = V - X[k,V] *** g_k.

k \ V	0		1		2		3		4		5		6		7		8		9	
1	0	0	0	0	0	0	4	1	4	1	4	1	8	2	8	2	8	2	12	3
2	0	0	0	0	0	0	4	0	5	1	5	1	8	0	9	1	10	2	12	0
3	0	0	0	0	0	0	4	0	5	0	6	1	8	0	9	0	10	0	12	0
4	0	0	0	0	3	1	4	0	6	2	7	1	9	3	10	2	12	4	13	3
5	0	0	1	1	3	0	4	0	6	0	7	0	9	0	10	0	12	0	13	0

- Ví dụ, trong bảng trên, ta sẽ xét các đồ vật từ 5 đến 1. Khởi đầu V = W = **9**.
  - Với k = 5, vì X[5, **9**] = 0 nên ta không chọn đồ vật loại 5 (**$x_5 = 0$**)
  - Với k = 4, vì X[4, **9**] = 3 nên ta chọn 3 đồ vật loại 4 (**$x_4 = 3$**) ⇒ V = 9 – 3 * 2 = **3**.
  - Với k = 3, vì X[3, **3**] = 0 nên ta không chọn đồ vật loại 3 (**$x_3 = 0$**)
  - Với k = 2, vì X[2, **3**] = 0 nên ta không chọn đồ vật loại 2 (**$x_2 = 0$**)
  - Với k = 1, vì X[1, **3**] = 1 nên ta chọn 1 đồ vật loại 1 (**$x_1 = 1$**) ⇒ V = 3 – 1 * 3 = **0**.

Vậy phương án **X=(1, 0, 0, 3, 0)** với TTL= **3 * 2 + 1 * 3 = 9**; TGT= **3 * 3 + 1 * 4 = 13**.

Bài tập: Bài toán MA TRẬN SỐ

Bài tập: Cho bảng **A** kích thước m x n (m dòng, n cột) ghi các số nguyên. Một người xuất phát từ cột đầu tiên (cột 1) cần sang cột cuối cùng (cột n). **Quy tắc di chuyển**: **từ ô (i, j) có thể di chuyển sang các ô (i-1, j+1), (i, j+1) và (i+1, j+1).** Hãy tìm **vị trí xuất phát** và **hành trình** đi từ cột 1 sang cột n sao cho **tổng giá trị** của các số ghi trên đường đi là **lớn nhất** ?

A	*1*	*2*	…	*n - 1*	*n*
1	a_{11}	a_{12}	…	a_{1n-1}	a_{1n}
2	a_{21}	a_{22}	…	(i-1, j+1)	a_{2n}
…	…	…	(i, j)	(i, j+1)	…
m	a_{m1}	a_{m2}	…	(i+1, j+1)	a_{mn}

→ Mô tả **kỹ thuật tham ăn** có thể áp dụng vào bài toán?

→ Đề xuất công thức truy hồi cho phép giải bài toán bằng **kỹ thuật quy hoạch động**?

Bài tập: Bài toán MA TRẬN SỐ

Ví dụ: Cho bảng **A** kích thước 4 x 5 (4 dòng, 5 cột) với các số nguyên như bên.

→ Hãy tìm **vị trí xuất phát** ở cột 1 và **hành trình** đi từ cột 1 sang cột n sao cho tổng giá trị của các số ghi trên đường đi là **lớn nhất** ?

f(X) → Max

A	*1*	*2*	*3*	*4*	*5*
1	1	2	6	7	6
2	7	8	7	5	6
3	8	5	1	6	5
4	4	11	10	9	2