ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

THIẾT KẾ VÀ ĐÁNH GIÁ THUẬT TOÁN

Bài 7
Phương pháp Quy hoạch động
(Dynamic Programing)

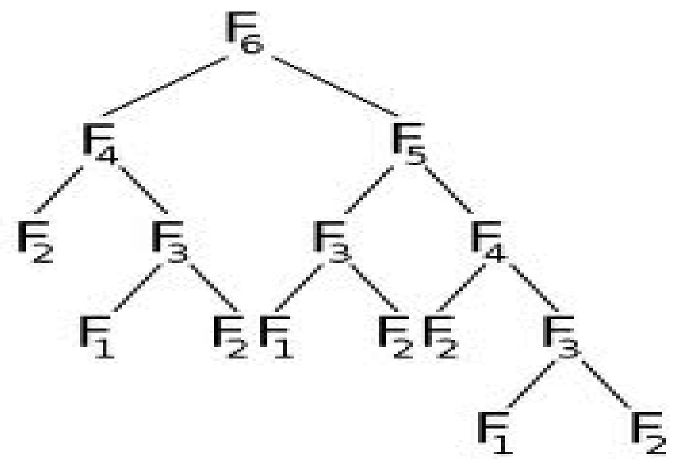
Nguyễn Thị Hồng Minh
minhnth@gmail.com

Nội dung

1. **Giới thiệu phương pháp**
2. **Các yếu tố cơ bản của phương pháp**
3. **Các bước giải bài toán theo phương pháp qui hoạch động**
4. **Một số bài toán cơ bản giải bằng qui hoạch động**
5. **Những lỗi hay mắc phải khi giải bài toán bằng phương pháp qui hoạch động**

Giới thiệu phương pháp

- **Xem lại bài toán Fibonacci**
  - Sơ đồ giải bài toán bằng đệ qui



    - Hạn chế: Nhiều bài toán con trùng nhau
    - Giải pháp khắc phục:
      - Giải tất cả các bài toán con, lưu vào bảng kết quả
      - Kết hợp kết quả các bài toán con => lời giải

$$F_1 ;\ F_2 \rightarrow F_3 \rightarrow \ldots \rightarrow F_n$$

Giới thiệu phương pháp

- **Ý tưởng phương pháp qui hoạch động**
  - Qui hoạch động (Dynamic Programing-DP) là phương pháp giải các bài toán bằng cách kết hợp lời giải của các bài toán con theo nguyên tắc:
    - Giải tất cả các bài toán con (một lần)
    - Lưu lời giải của các bài toán vào một bảng
    - Phối hợp các bài toán con để nhận lời giải bài toán ban đầu.
  - Cách phát biểu khác: 1 bài toán giải bằng QHĐ được phân rã thành các bài toán con và bài toán lớn hơn sẽ được giải quyết thông qua các bài toán con này (bằng các phép truy hồi).
  - Phương pháp QHĐ thường dùng cho các bài toán tối ưu.

Giới thiệu phương pháp

- **So sánh đệ qui và QHĐ**
  - ***Đệ qui:***
    - Coi như các bài toán con đã có lời giải, qui bài toán lớn về các bài toán nhỏ hơn, cho tới khi gặp trường hợp suy biến.
    - Lời giải cho cùng 1 bài toán có thể bị lặp lại.
    - Phương thức Top-Down
  - ***Qui hoạch động:***
    - Giải trước tất cả các bài toán con rồi giải bài toán lớn dần
    - Kết quả của bài toán con (tối ưu) được lưu vào bảng phương án nên không bị giải lặp lại khi cần kết quả.
    - Phương thức Bottom-Up

Giới thiệu phương pháp

- **Nhận dạng bài toán giải bằng QHĐ**

  Các bài toán có các đặc trưng sau thì có thể giải bằng QHĐ

  - Bài toán có thể giải bằng đệ qui và tìm lời giải tối ưu.
  - Bài toán có thể phân rã thành nhiều bài toán con mà sự phối hợp lời giải của các bài toán con sẽ cho ta lời giải của bài toán ban đầu.
  - Quá trình tìm ra lời giải của bài toán ban đầu từ các bài toán con đơn giản hơn được thực hiện qua một số hữu hạn các bước có tính truy hồi.

Giới thiệu phương pháp

- **Điều kiện để thiết kế và thực hiện giải thuật QHĐ**
  - Câu hỏi: “Có phải mọi bài toán tối ưu đều có thể giải bằng phương pháp QHĐ hay không? ”
  - Câu trả lời: “Nếu tìm được công thức truy hồi để giải bài toán lớn từ các bài toán nhỏ hơn thì sẽ giải bài toán được bằng QHĐ.”

  => ***Vấn đề tìm công thức truy hồi là vấn đề quan trọng nhất khi giải một bài toán bằng phương pháp QHĐ.***
  - Hơn nữa: Đủ không gian bộ nhớ cần thiết để lưu kết quả các bài toán con trong quá trình truy hồi.

Các yếu tố cơ bản của phương pháp

- **Công thức truy hồi (công thức QHĐ)**
- **Cơ sở QHĐ**
- **Bảng phương án**
- **Kết quả tối ưu của bài toán**
- **Truy vết tìm nghiệm**

Các yếu tố cơ bản của phương pháp

- **Công thức truy hồi**
  - Công thức xác định lời giải bài toán lớn hơn thông qua các bài toán nhỏ hơn
  - Chia bài toán thành N giai đoạn, tối ưu dần từng giai đoạn từ 0, 1, ..i,.., N.
  - Giai đoạn thứ i: Gọi $F(i)$ là hàm số xác định giá trị tối ưu (max/min) cần xác định. Khi đã biết các giá trị tối ưu (max/min) $F(j)$ của các giai đoạn j=0,..i-1 thì:

    F(i)=Min {F(j) *kết hợp với* ***Giá trị nhỏ nhất từ j đến i} với j=0,1,2,..,i-1***

    Hoặc

    F(i)=Max {F(j) *kết hợp với* ***Giá trị lớn nhất từ j đến i) với j=0,1,2...,i-1***

$$F(N) \longleftarrow F(k)\ \forall k<N \longleftarrow F(j)\ \forall j<k \longleftarrow \ldots \longleftarrow F(2), F(1), F(0)$$

Các yếu tố cơ bản của phương pháp

- **Cơ sở QHĐ**

$$F(N) \longleftarrow F(k)\ \forall k<N \longleftarrow F(i)\ \forall i<k \longleftarrow \ldots \longleftarrow F(2), F(1), F(0)$$

  - Là tất cả các giá trị tối ưu có thể tính được ngay $F(2)$, $F(1)$, $F(0)$
- **Bảng phương án**
  - Bảng lưu tất cả các giá trị tối ưu $F(i)$ $(i=0,1,\ldots,N)$ của các bài toán con.
  - Bảng phương án có thể là 1 chiều (mảng), 2 chiều (ma trận) hoặc nhiều hơn
  - Chú ý cách phối hợp kết quả của các bài toán để tìm kết quả cho bài toán lớn hơn (dựa vào công thức truy hồi)
- **Kết quả tối ưu của bài toán**
  - Nằm trong bảng phương án
  - Vị trí: Tùy thuộc bài toán

Các yếu tố cơ bản của phương pháp

- **Ví dụ xây dựng công thức truy hồi và bảng phương án**
  - ***Bài toán dãy con đơn điệu tăng dài nhất :*** Tìm dãy con dài nhất của một dãy đã cho. Các phần tử có thể không liên tiếp.

    Ví dụ: Dãy 2; 5; 4; 6; 3; 8; 9; 7

    => Dãy con dài nhất có 5 phần tử: 2; 5; 6; 8; 9

    Phân tích:

    - Input: (a,n)
    - Output: m – số lớn nhất các phần tử của dãy theo thứ tự tăng dần
    - Hàm tối ưu $L(i)$: Độ dài dãy con đơn điệu tăng dài nhất đến phần tử i

      Là độ dài các dãy con dài nhất đến j cộng 1 khi ghép thêm a_i vào sau, với điều kiện $j<i$, $a_j<a_i$
    - Công thức truy hồi: $L(i) = \max\{L(j)\}+1$ với $j<i$, $a_j<a_i$
    - Cơ sở QHĐ: $L(0) = 0$; $L(1) = 1$

Các yếu tố cơ bản của phương pháp

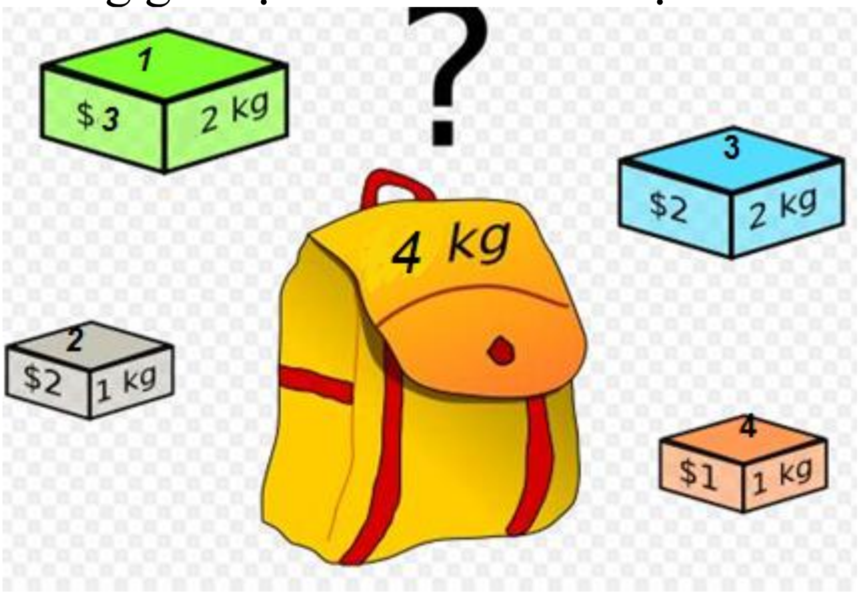
- **Ví dụ xây dựng công thức truy hồi và bảng phương án**
  - ***Bài toán dãy con đơn điệu tăng dài nhất***
    - Bảng phương án: Bảng 1 chiều L

i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
a_i		2	5	4	6	3	8	9	7
L(i)	0	1	2	2	3	2	4	5	4

    - Giá trị tối ưu: $\max(L(i))_{i=1..n} = 5$
    - Vậy dãy con đơn điệu tăng dài nhất có độ dài là 5
    - Vấn đề đặt ra: Dãy nào?
      - Từ giá trị tối ưu của bài toán tìm được, sử dụng bảng phương án để tìm nghiệm của bài toán => Truy vết tìm nghiệm

Các yếu tố cơ bản của phương pháp

- **Ví dụ xây dựng công thức truy hồi và bảng phương án**
  - ***Bài toán xếp balo 0-1:*** có *N* đồ vật với trọng lượng và giá trị tương ứng (p_i,w_i). Tìm cách cho các vật vào balo có trọng lượng *P* sao cho đạt giá trị cao nhất. Mỗi vật chỉ được chọn 1 lần (0-1).
    - Input: n, (p_i,w_i) i=1..n, P
    - Output: W= tổng giá trị lớn nhất các đồ vật cho vào balo



Các yếu tố cơ bản của phương pháp

- **Ví dụ xây dựng công thức truy hồi và bảng phương án**
  - ***Bài toán xếp balo 0-1:***
    - o Hàm tối ưu $C(i,j)$: Giá trị lớn nhất khi chọn các vật từ 1 đến i với trọng lượng balo j (i=0..n, j=0..W).

      - Nếu không chọn đồ vật thứ i thì: $C(i,j)= C(i\text{-}1,j)$

    - Nếu có chọn đồ vật thứ i thì $C(i,j) = C(i\text{-}1,j\text{-}p_i)+w_i$ (đkiện $p_i<=j$)
    - o Công thức truy hồi

$$C(i,j) = \max(C(i\text{-}1,j), C(i\text{-}1,j\text{-}p_i)+w_i)$$

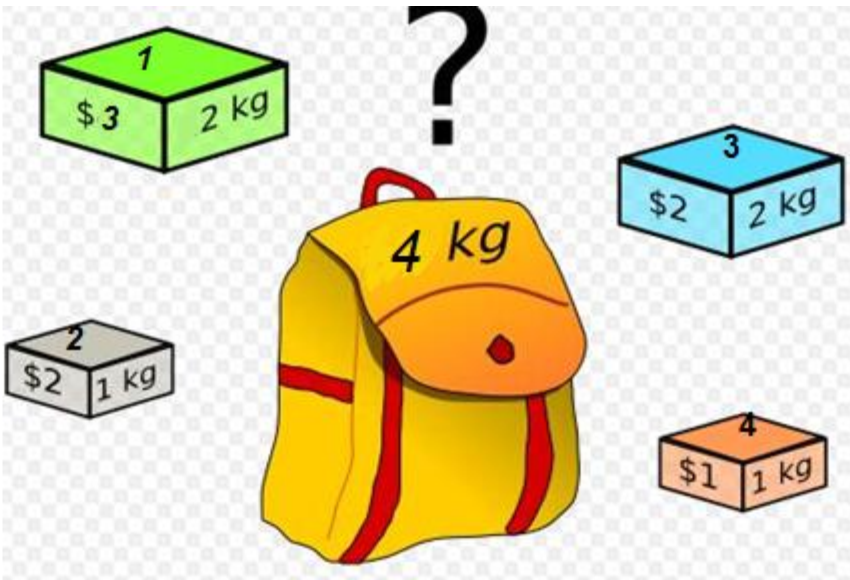
    - o Cơ sở QHD

$$C(i,0) = 0$$

$$C(0,j) = 0$$

Các yếu tố cơ bản của phương pháp

- **Ví dụ xây dựng công thức truy hồi và bảng phương án**
  - ***Bài toán xếp balo 0-1:***
    - o Bảng phương án: Bảng hai chiều lưu các giá trị $C(i,j)$



i \ j	0	1	2	3	4
0	0	0	0	0	0
1	0	0	3	3	3
2	0	2	3	5	5
3	0	2	3	5	5
4	0	2	3	5	6

$C(i,j) = \max(C(i\text{-}1,j),\ C(i\text{-}1,j\text{-}p_i)+w_i)$

- o Giá trị tối ưu: $C(n,W) = 6$
- o Những vật nào sẽ cho vào balo? => Truy vết tìm nghiệm

Các yếu tố cơ bản của phương pháp

- **Truy vết tìm nghiệm**
  - Dựa vào bảng phương án để dần dần tìm ra các thành phần nghiệm của bài toán.
  - Điểm bắt đầu của quá trình truy vết: ô chứa giá trị tối ưu, kết quả của bài toán.
  - Điểm kết thúc của quá trình truy vết: tùy thuộc bài toán, thường nằm ở những dòng, những cột đầu tiên của bảng phương án.
  - ***Chú ý:*** cùng 1 bài toán có thể có nhiều lời giải cùng cho 1 kết quả tối ưu. Việc truy vết bình thường chỉ đưa ra 1 phương án. Để hiển thị toàn bộ các nghiệm thì có thể dùng giải thuật đệ qui quay lui khi truy vết.

Các yếu tố cơ bản của phương pháp

- **Truy vết tìm nghiệm**
  - ***Bài toán dãy con đơn điệu tăng dài nhất***

i	0	1	2	3	4	5	6	7	8
a_i		2	5	4	6	3	8	9	7
L(i)	0	1	2	2	3	2	4	5	4

- Bắt đầu từ vị trí tối ưu. $L(i)$ đạt max.
- Tại mỗi bước: Xét ô $L(i)$, phần tử đứng trước a_i trong dãy con là phần tử a_j nếu $j<i$, $a_j<a_i$ và $L(j)=L(i)-1$.
- Kết thúc khi đến phần tử đầu tiên của dãy con ($L(i) = 1$)
- Chú ý: Có thể có nhiều giá trị cùng thỏa mãn điều kiện này nhưng ta chỉ chọn 1 trong các phương án đó.

Các yếu tố cơ bản của phương pháp

- **Truy vết tìm nghiệm**
  - ***Bài toán xếp balo 0-1***

$i \backslash j$	0	1	2	3	4
0	0	0	0	0	0
1	0	0	**3**	3	3
2	0	2	3	**5**	5
3	0	2	3	**5**	5
4	0	2	3	5	**6**

i	p_i	w_i	Chọn
0			
1	2	3	✓
2	1	2	✓
3	2	2	x
4	1	1	✓

- Bắt đầu từ ô $C(n,W)$
  - Nếu $C(i,j)= C(i-1,j)$ thì vật thứ *i* không được chọn và xét $C(i-1,j)$
  - Ngược lại $C(i,j) = C(i-1,j-p_i)+w_i$ thì vật thứ *i* được chọn và xét tiếp ô $C(i-1,j-p_i)$
- Kết thúc khi tới hàng 0 hoặc cột 0

Các bước giải bài toán bằng QHĐ

- **Các bước cơ bản để giải bài toán bằng QHĐ**

1.Nhận dạng bài toán giải bằng QHĐ

2.Xây dựng công thức truy hồi

3.Xác định và xây dựng cơ sở QHĐ

4.Dựng bảng phương án

5.Tìm kết quả tối ưu

6.Truy vết liệt kê thành phần nghiệm

Các bước giải bài toán bằng QHĐ

- **Nhận dạng bài toán giải bằng QHĐ**
  - Dựa vào dấu hiệu nhận dạng bài toán giải bằng QHĐ
  - Chú ý: không phải bài toán nào giải theo QHĐ cũng tốt hơn đệ qui, chẳng hạn như bài toán "Xếp ba lô 0-1" với số liệu lớn.
- **Xây dựng công thức truy hồi**
  - Đưa bài toán về 1 dạng cơ bản, và triển khai ý tưởng của dạng bài toán đó để nhanh chóng nhận ra hướng thiết lập công thức truy hồi.
  - Đây là bước khó nhất và cũng quan trọng nhất trong toàn bộ quá trình thiết kế thuật toán cho bài toán.
- **Xác định cơ sở QHĐ**
  - Dựa vào công thức truy hồi để nhận ra các bài toán cơ sở.
  - Dựa vào ý nghĩa của công thức truy hồi để thiết lập giá trị cho cơ sở.

Các bước giải bài toán bằng QHĐ

- **Dựng bảng phương án**
  - Dựa vào công thức truy hồi để tính giá trị các ô trong bảng phương án.
  - Chú ý: bảng phương án có thể 1 chiều, 2 chiều hoặc nhiều hơn
- **Tìm kết quả tối ưu**
  - Xác định vị trí chứa kết quả tối ưu của bài toán trên bảng phương án.
  - Chú ý: ngoài kết quả tối ưu, ô chứa kết quả tối ưu còn là điểm bắt đầu cho quá trình truy vết tìm nghiệm => lưu tọa độ của ô đó.
- **Truy vết liệt kê thành phần nghiệm**
  - Từ điểm bắt đầu là vị trí chứa kết quả tối ưu
  - Truy ngược lại về điểm bắt đầu của nghiệm: có thể là những ô đầu tiên trong bảng phương án (bài toán cơ sở), có thể là ô của bảng phương án đạt giá trị đầu.

Một số bài toán

- **Bài toán *Đường đi trong bảng số*:** Cho bảng số dương kích thước *m*x*n*. Từ 1 ô (*i*, *j*) có thể đi tới các ô phía bên phải (*i*-1, *j*+1), (*i*, *j*+1), (*i*+1, *j*+1). Tìm cách đi từ bên trái sang ô (*m*, *n*) sao cho tổng giá trị các ô trên đường đi là lớn nhất.

  - Input: Ma trận (*a*,*m*,*n*)
  - Output:
    - Tổng giá trị lớn nhất các ô đi từ (*,1) đến (*m*,*n*)
    - Các ô trên đường đi.
  - Ví dụ:

    Đường đi dài nhất từ (1,1) đến (6,7)

9	9	9	2	6	1	6	1
4	**5**	2	1	4	5	1	2
2	3	**9**	**8**	1	3	4	3
5	7	2	1	**9**	8	1	4
9	8	2	1	4	**7**	6	5
3	7	1	3	6	8	**1**	6
1	2	3	4	5	6	7	

Một số bài toán

- **Bài toán *Đường đi trong bảng số*:**
  - **Nhận dạng bài toán giải bằng QHĐ**
    - Tìm lời giải tối ưu.
    - Có thể phân rã thành nhiều bài toán con, phối hợp cho lời giải bài toán ban đầu.
    - Các bước có tính truy hồi.
  - **Xây dựng công thức truy hồi**
    - Hàm tối ưu: $C(i,j)$ là tổng giá trị các ô trên đường đi tới ô (i, j)
    - Để đến được ô (i, j), có thể từ 1 trong 3 ô $(i-1,j-1)$, $(i,j-1)$, $(i+1,j-1)$

      Vậy $C(i,j) = \max\{C(i-1,j-1), C(i,j-1), C(i+1,j-1)\} + a_{ij}$

    Chú ý là các ô được xét phải nằm trong bảng số.
  - **Cơ sở QHĐ**

    $C(i,0) = C(0,j)=0$

Một số bài toán

- **Bài toán *Đường đi trong bảng số*:**
  - **Tìm kết quả tối ưu**

    $C(m,n)$ là tổng giá trị các ô khi đi từ bên trái tới ô (m, n)

  - **Truy vết**
    - o Bắt đầu từ ô (m,n), kết thúc khi tới cột đầu tiên
    - o Xét tại ô (i, j), xem 1 trong 3 ô C(i-1,j-1),C(i,j-1),C(i+1,j-1), ô nào có giá trị = $C(i,j)$ - a_{ij} thì (i,j) được tới từ ô đó, tiếp tục truy vết ngược lại.

Một số bài toán

- **Bài toán *Đường đi trong bảng số*:**
  - **Truy vết**

9	9	9	2	6	1	6
4	5	2	1	4	5	1
2	3	**9**	**8**	1	3	4
5	7	2	1	**9**	8	1
9	8	2	1	4	7	6
3	7	1	3	6	8	**1**

Bảng số a_{mn}

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	0	0	0	0	0
1	0	9	18	27	29	35	38	48
2	0	4	14	20	28	37	42	46
3	0	2	8	**25**	**33**	34	45	54
4	0	5	**16**	19	26	**42**	50	51
5	0	**9**	17	19	20	30	**49**	56
6	0	3	16	18	22	28	38	**50**

Bảng phương án C_{mn}

Một số bài toán

- **Bài toán *Đường đi trong bảng số*:**
  - **Lược đồ chương trình**

```
/*Cơ sở QHĐ*/
for (i=0…m) C[i][0] = 0
for (j=0…n) C[0][j] = 0

/*Xây dựng bảng phương án*/
for (i=1…m)
   for (j=1…n)
     C[i][j] = max(C[i-1][j-1] ,C[i][j-1] ,C[i+1][j-1])+A[i][j]

/*Tổng giá trị các ô trên đường đi*/
print C[m][n]
```

Một số bài toán

- **Bài toán *Đường đi trong bảng số*:**
  - **Lược đồ chương trình**

```
/*Truy vết*/
i=m; j=n;
while(j>1)
    print (i, j)
    if(i>1)
        k = C[i][j]-A[i][j];
        if(C[i-1][j-1]=k) {i--; j--;}
        else if(C[i][j-1]=k) j--;
            else i--;
    else j--;
endw;
```

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	9	18	27	29	35	38	48
2	0	4	14	20	28	37	42	46
3	0	2	8	**25**	**33**	34	45	54
4	0	5	**16**	19	26	**42**	50	51
5	0	**9**	17	19	20	30	**49**	56
6	0	3	16	18	22	28	38	**50**

Một số lỗi thường gặp

- **Lỗi trong khi thiết kế thuật toán**
  - **Nhận dạng không đúng bài toán**
  - **Xác định sai công thức truy hồi**
    - Xác định sai giá trị tối ưu cần tính
    - Nhầm lẫn giữa các cách kết hợp bài toán con
    - Xác định sai ý nghĩa của bảng phương án
  - **Xác định sai hoặc thiếu cơ sở QHĐ**
  - **Xác định sai kết quả tối ưu**
    - Muốn xác định được kết quả cần tìm nằm ở đâu trong bảng phương án phải dựa vào ý nghĩa của công thức truy hồi.
  - **Xác định sai điều kiện dừng khi truy vết**
    - Quá trình truy vết bắt đầu tại ô chứa kết quả tối ưu, nhưng kết thúc tại đâu còn phải dựa vào công thức truy hồi.

Một số lỗi thường gặp

- **Lỗi trong khi lập trình**
  - **Kích thước bảng phương án nhỏ hơn yêu cầu**
    - Bảng phương án có thể có thêm viền (hàng số 0, cột số 0) => bảng phương án cũng lớn hơn hoặc bằng kích thước miền dữ liệu.
  - **Không phải bảng phương án nào cũng tính theo từng hàng**
    - Việc hình thành dần kết quả của các bài toán con và lưu vào bảng phương án phải được thực hiện tuần tự nhưng tùy vào công thức truy hồi thì mới có thể biết được là tính bảng phương án theo từng hàng hay theo từng cột.
  - **Xét ô ở ngoài bảng**
    - Khi tính bảng phương án hoặc truy vết, cần chú ý các chỉ số để đảm bảo phần tử ở trong bảng.

Một số dạng bài toán QHĐ cơ bản

1. **Bài toán "Dãy con đơn điệu tăng dài nhất"**
2. **Bài toán "Xếp ba lô 0-1"**
3. **Bài toán "Đường đi trong bảng số"**
4. **Bài toán "Dãy con tổng bằng S"**

   Cho 1 tập có N số nguyên dương. Tìm 1 tập con có tổng bằng giá trị dương cho trước S.
5. **Bài toán "Biến đổi xâu"**

   Với 3 phép biến đổi xâu như sau:

   Chèn 1 kí tự vào sau vị trí i

   Xóa kí tự tại vị trí i

   Thay thế kí tự tại vị trí i bằng 1 kí tự khác

   Cho trước 2 xâu X, Y. Tìm cách biến đổi X về Y sao cho số phép biến đổi là ít nhất.

   Ví dụ: X = "aec"; Y = "bcd" Sử dụng ít nhất 3 phép biến đổi. Một cách là:

   - Chèn d vào sau vị trí thứ 3 (sau c); X="aecd"
   - Xóa kí tự ở vị trí thứ 2 (e); X="acd"
   - Thay kí tự ở vị trí số 1 (a) bằng kí tự b X="bcd"

Đưa một số bài toán về dạng cơ bản

1. **Bài toán "Dãy con chung dài nhất"**

   Cho 2 dãy số X (M phần tử) và Y(N phần tử)

   Tìm dãy con chung dài nhất của 2 dãy này

   - Dạng cơ bản: Biến đổi xâu

2. **Bài toán "Chia kẹo"**

   Cho N gói kẹo. Gói kẹo thứ i có A[i] cái kẹo.

   Chia các gọi kẹo thành 2 phần để chênh lệch giữa 2 phần là ít nhất

   - Dạng cơ bản: Dãy con tổng bằng S

3. **Bài toán "Lập lịch sử dụng tài nguyên"**

   - Dạng cơ bản: Dãy con đơn điệu tăng dài nhất