VANLANG UNIVERSITY

School of Technology
Information Technology

ACM – ICPC Programming Contest

GV: Võ Anh Tiến
Đại học Văn Lang

VANLANG UNIVERSITY
School of Technology
Information Technology

Quy hoạch động

- Là kỹ thuật để giải các bài toán được định nghĩa đệ quy theo cách không đệ quy.
- Được phát minh bởi nhà toán học người Mỹ: Richard Bellman 1950.
- Ý tưởng:
  - Lưu lại các trị của các lần tính toán trước làm dữ liệu cho việc tính toán của lần sau (kết hợp nghiệm các bài toán cỡ nhỏ ta thu được nghiệm bài toán cỡ lớn).
  - Đi từ biên (trường hợp riêng đơn giản nhất), đi tới điểm kết thúc (tìm ra nghiệm bài toán).
- Các bước thực hiện
  - Mô hình hóa bài toán, thành lập công thức đệ quy
  - Lập bảng thể hiện giải pháp bài toán
  - Rút ra giải pháp từ bảng được lập

Quy hoạch động

- Các bài toán
  - Bài toán người du lịch Manhattan
  - Bài toán leo núi
  - Chuỗi con chung liên tiếp dài nhất
  - Chuỗi con chung không liên tiếp dài nhất
  - Bài toán cái túi
  - Bài toán lập lịch có trọng số
  - Bài toán đặt trạm đổi tiền

VANLANG UNIVERSITY
School of Technology
Information Technology

Quy hoạch động

- **Mô hình hóa**
- Giai thừa
  - n! = n* (n-1)!
  - n = 1 với n=0
- **Lập bảng**

n!	1	1	2	6	24	120
n	0	1	2	3	4	5

- **Rút ra giải pháp từ bảng**

```
long giaiThua( int n){
        long gt=1;
    for (int i =2; i<=n;i++) gt=gt*i;
    return gt;
}
```

- **Mô hình hóa**
- Fibonacii
  - F(n) = F(n-1) + F(n-2) với n≥2
  - F(0) = 0
  - F(1) = 1
- **Lập bảng**

F(n)	0	1	1	2	3	5
n	0	1	2	3	4	5

- **Rút ra giải pháp từ bảng**

```
long fibo(int n){
        long* f=new long[n+1];
        f[0]=0;f[1]=1;
        for (int i=2;i<=n;i++)
            f[i]=f[i-1] + f[i-2];
        return f[n];
}
```

Hệ số nhị thức

- Hệ số nhị thức

$$C\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad for\, 0 \le k \le n$$

- Công thức nhị thức

$$(a+b)^n = C(n,0)a^n + \ldots + C(n,i)a^{n-i}b^i + \ldots + C(n,n)b^n$$

$$C\binom{n}{k} = \begin{cases} C\binom{n-1}{k-1} + C\binom{n-1}{k} & 0 < k < n \\ 1 & k = 0 \;\; or \;\; k = n \;\; (C\binom{n}{0} \;\; or \;\; C\binom{n}{n}) \end{cases}$$

Hệ số nhị thức

❖ Lập bảng n+1 dòng, k+1 cột

k / n	0	1	2	3	…	k-1	k
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
...							
k	1						1
…							
n-1	1					$C\binom{n-1}{k-1}$	$C\binom{n-1}{k}$
n	1						$C\binom{n}{k}$

Hệ số nhị thức

- Rút ra giải pháp từ bảng

Thuật toán C(*n*,*k*)

```
//Tính C(n, k) bằng giải thuật quy hoạch động
//Đầu vào: cặp số nguyên dương n ≥ k ≥ 0
//Đầu ra: giá trị C(n ,k)
for i← 0 to n do
    for j ←0 to min (i ,k) do
        if j = 0 or j = i
            C [i , j] ← 1
        else C [i , j] ← C[i-1, j-1] + C[i-1, j]
return C [n, k]
```

Hệ số nhị thức

- Cài đặt bằng C/C++

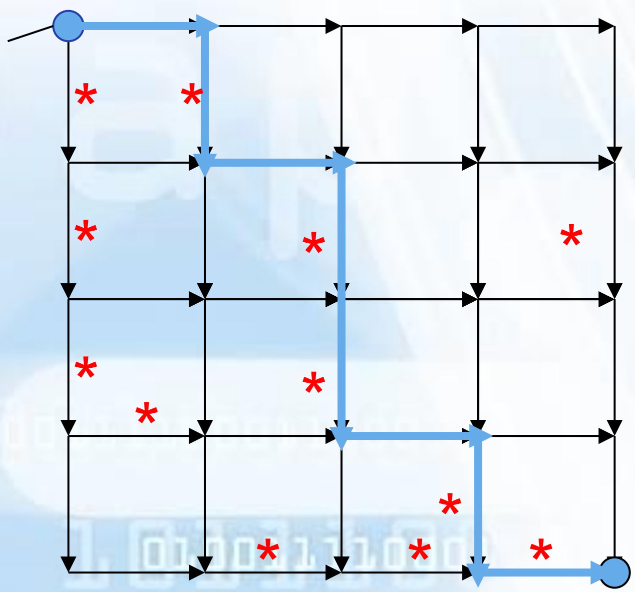
```
long C(int n,int k){
      //Tính C(n, k) bằng giải thuật quy hoạch động
      //Đầu vào: cặp số nguyên dương n ≥ k ≥ 0
      //Đầu ra: giá trị C(n ,k)
    If (k==0 || n==k) return 1;
    If (k>n/2) k=n-k;
    int min;
    for(int i=0;i<n;i++){
        min = i>k?k:i;              //chỉ cần tính tới k
        for(int j=0;j<min;j++)
            if (j == 0 || j == k)
                C [i ][ j] =1;
            else
                C [i][j]= C[i-1][j-1] + C[i-1][j];
    }
    return C [n][k];
}
```

VANLANG UNIVERSITY
School of Technology
Information Technology

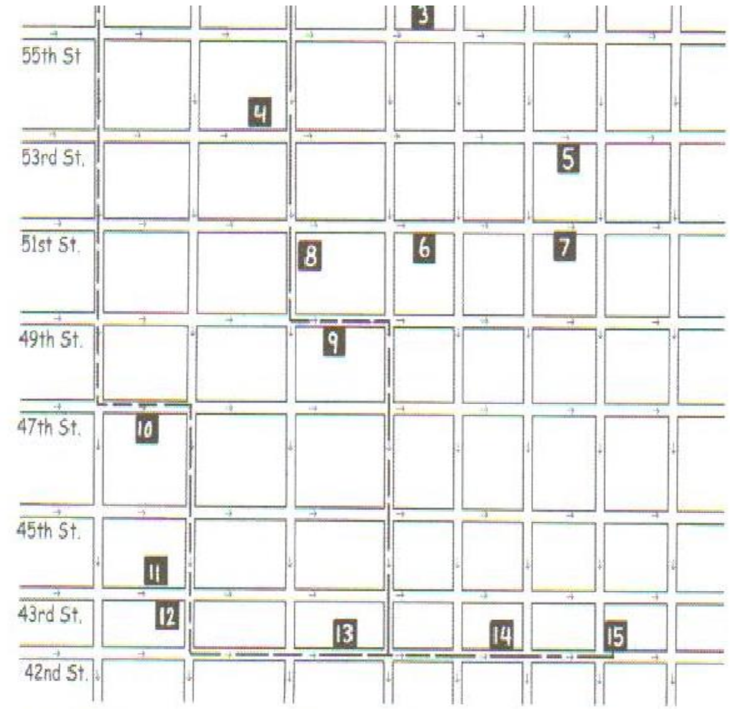
Bài toán người du lịch Manhattan

- Cần tìm một lối đi từ nơi xuất phát tới điểm dừng chân cuối cùng sao cho qua được nhiều địa điểm tham quan(dấu *) nhất. Người du lịch chỉ đi về phía đông hoặc phía nam.

Điểm xuất phát



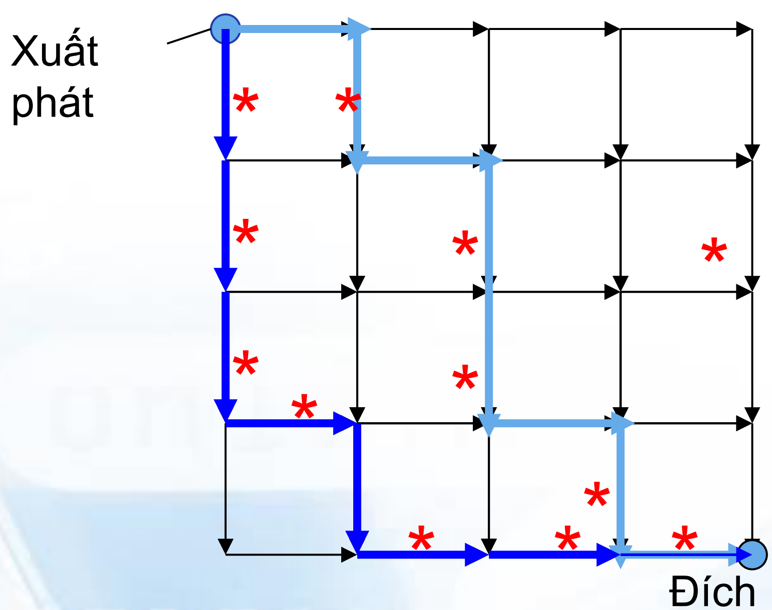
Điểm dừng cuối



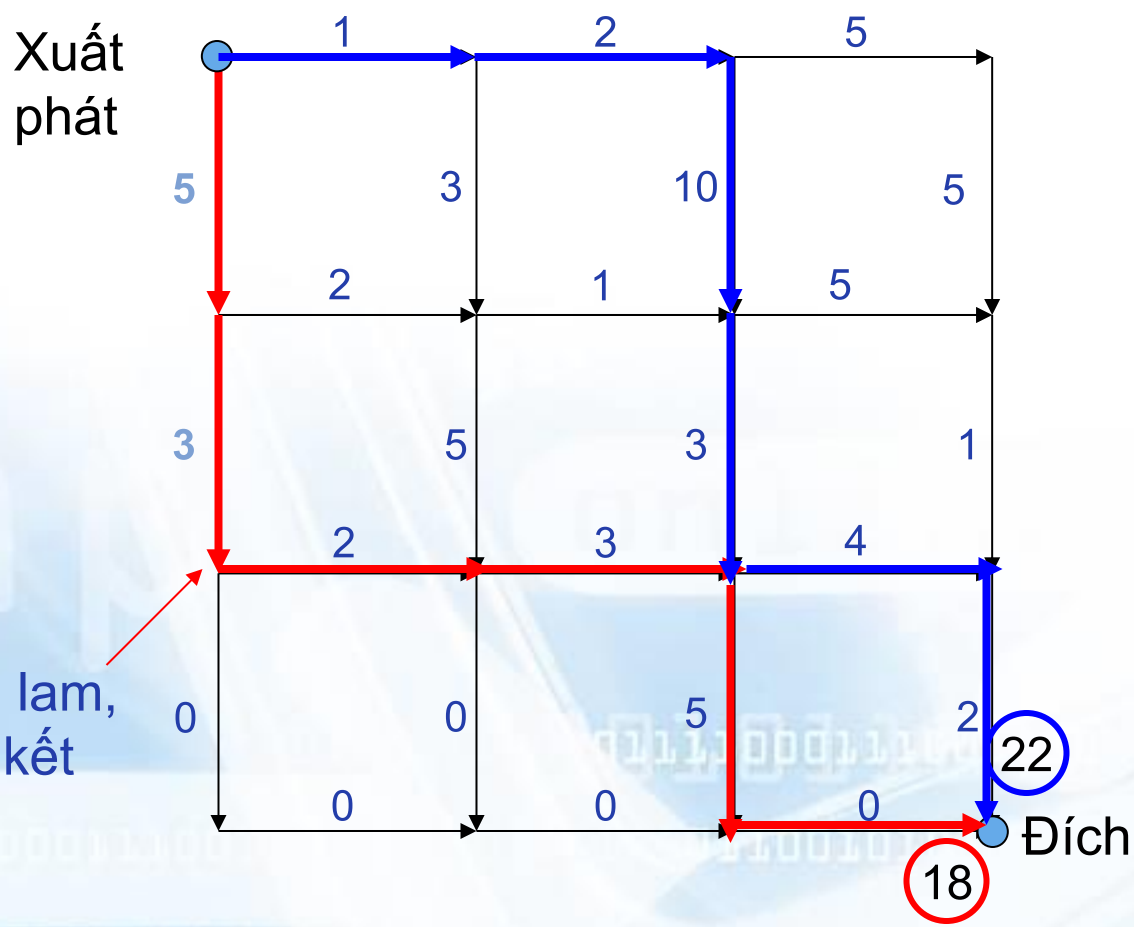
1 Carnegie Hall
2 Tiffany & Co.
3 Sony Building
4 Museum of Modern Art
5 Four Seasons
6 St. Patrick's Cathedral
7 General Electric Building
8 Radio City Music Hall
9 The Today Show
10 Paramount Building
11 NY Times Building
12 Times Square
13 General Society of Mechanics and Tradesmen (a must see!)
14 Grand Central Terminal
15 Chrysler Building

Bài toán người du lịch Manhattan

- Mục đích
  - Tìm đường đi có trọng số lớn nhất
- Đầu vào
  - Lưới G cho biết trọng số giữa hai đỉnh lưu trong mảng W.
- Đầu ra
  - Đường đi có trọng số lớn nhất trong G



Bài toán người du lịch Manhattan



Chọn tham lam, không cho kết quả tốt

Bài toán người du lịch Manhattan

- Mô hình hóa và lập công thức đệ quy
- SoDD(n,m)

  If (n=0 và m=0) return 0;

  if (*n=0)*

  return SoDD*(0,m-1) + Số địa điểm giữa nút (0,m-1) và (0,m)*

  if (m=*0)*

  return SoDD*(n-1,0) + Số địa điểm giữa nút (n-1,0) và nút (n,0)*

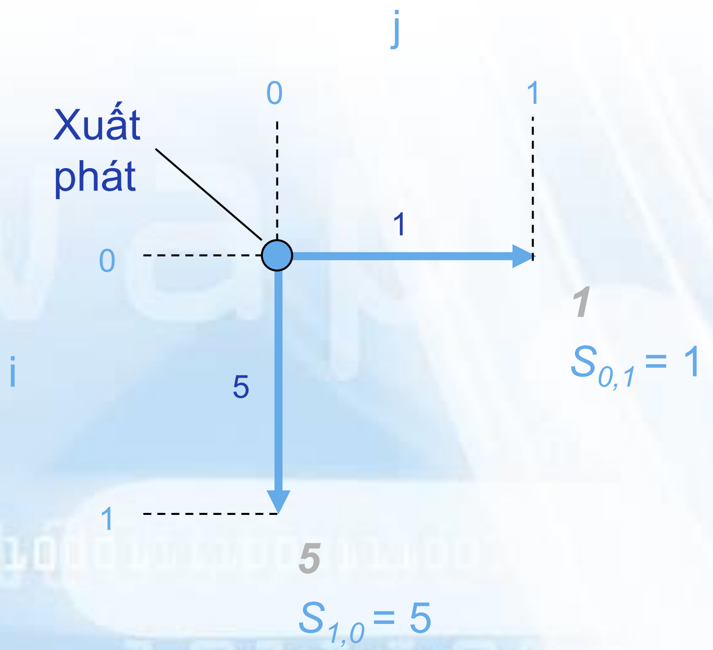
  x ← SoDD(n-1,m)+ Số địa điểm *giữa nút (n- 1,m) và nút (n,m)*

  y ← SoDD(n,m-1)+ *Số địa điểm giữa nút (n,m-1) và nút (n,m)*

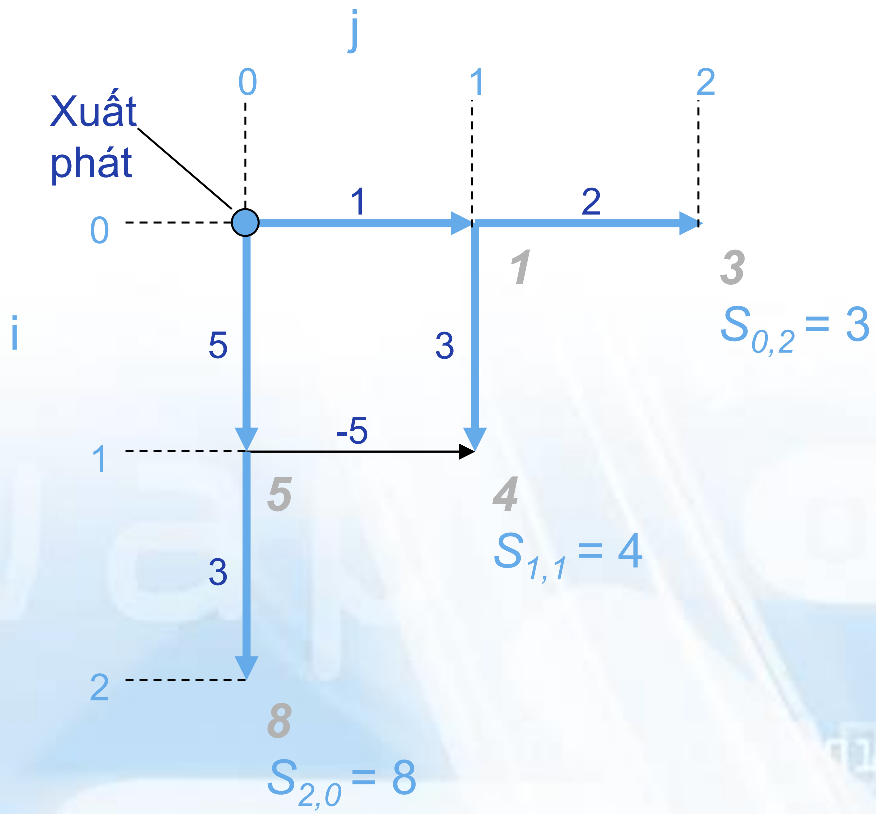
  return *max{x,y}*

Bài toán người du lịch Manhattan

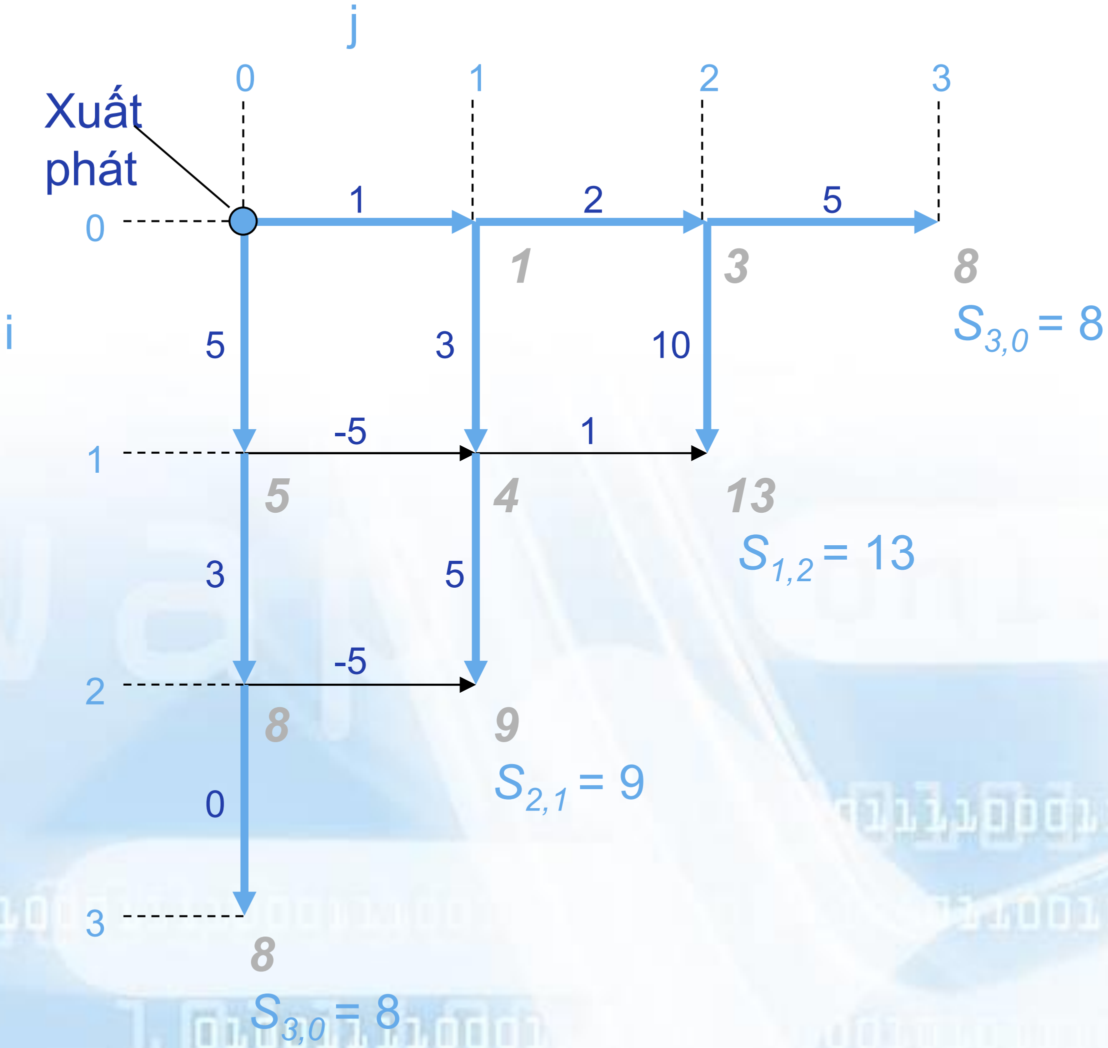
- Lập bảng
- Tính toán lối đi tối ưu cho mỗi đỉnh của đồ thị
- Giá trị tại mỗi đỉnh là giá trị tối đa của hai đỉnh trước đó.



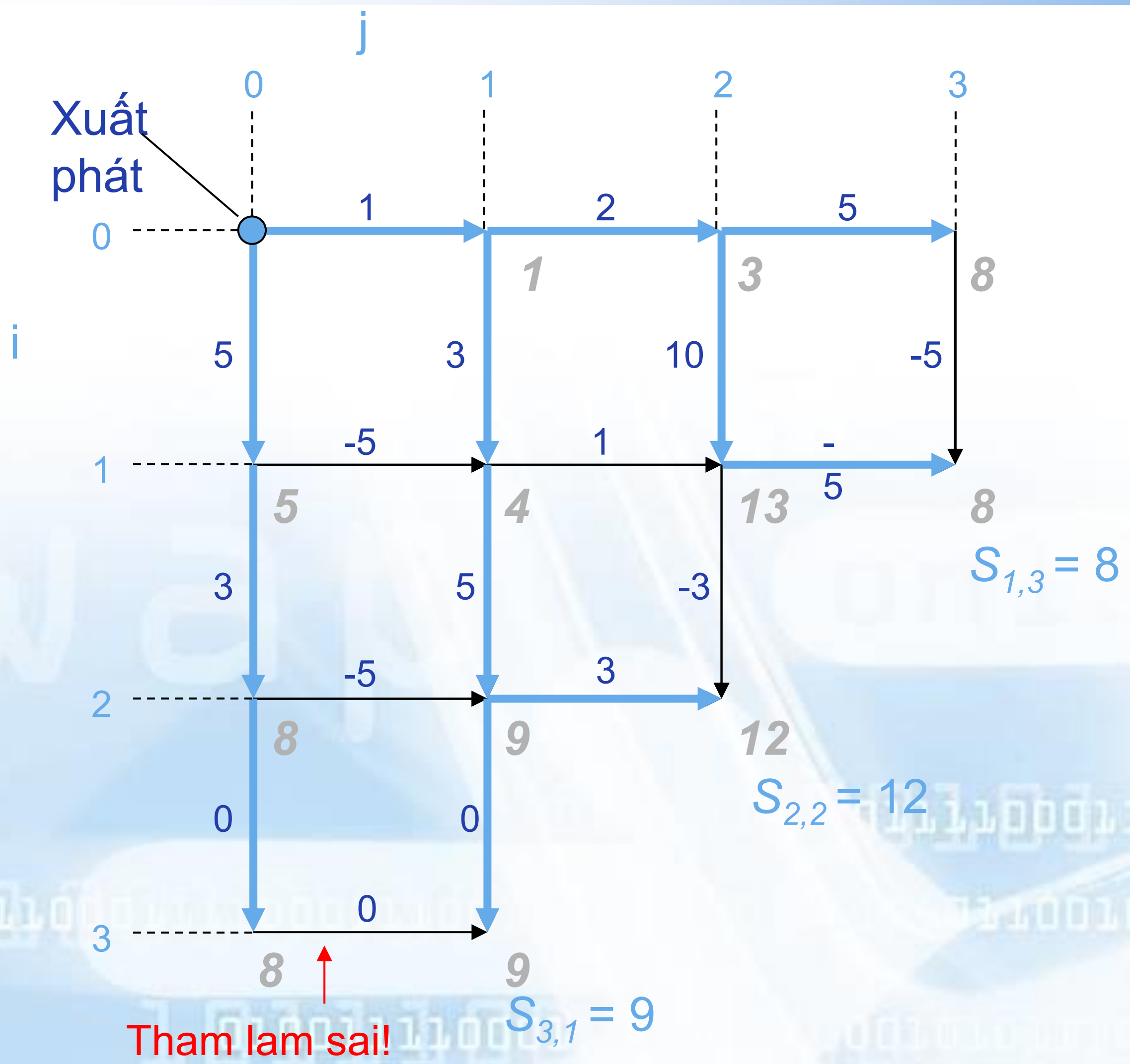
Bài toán người du lịch Manhattan



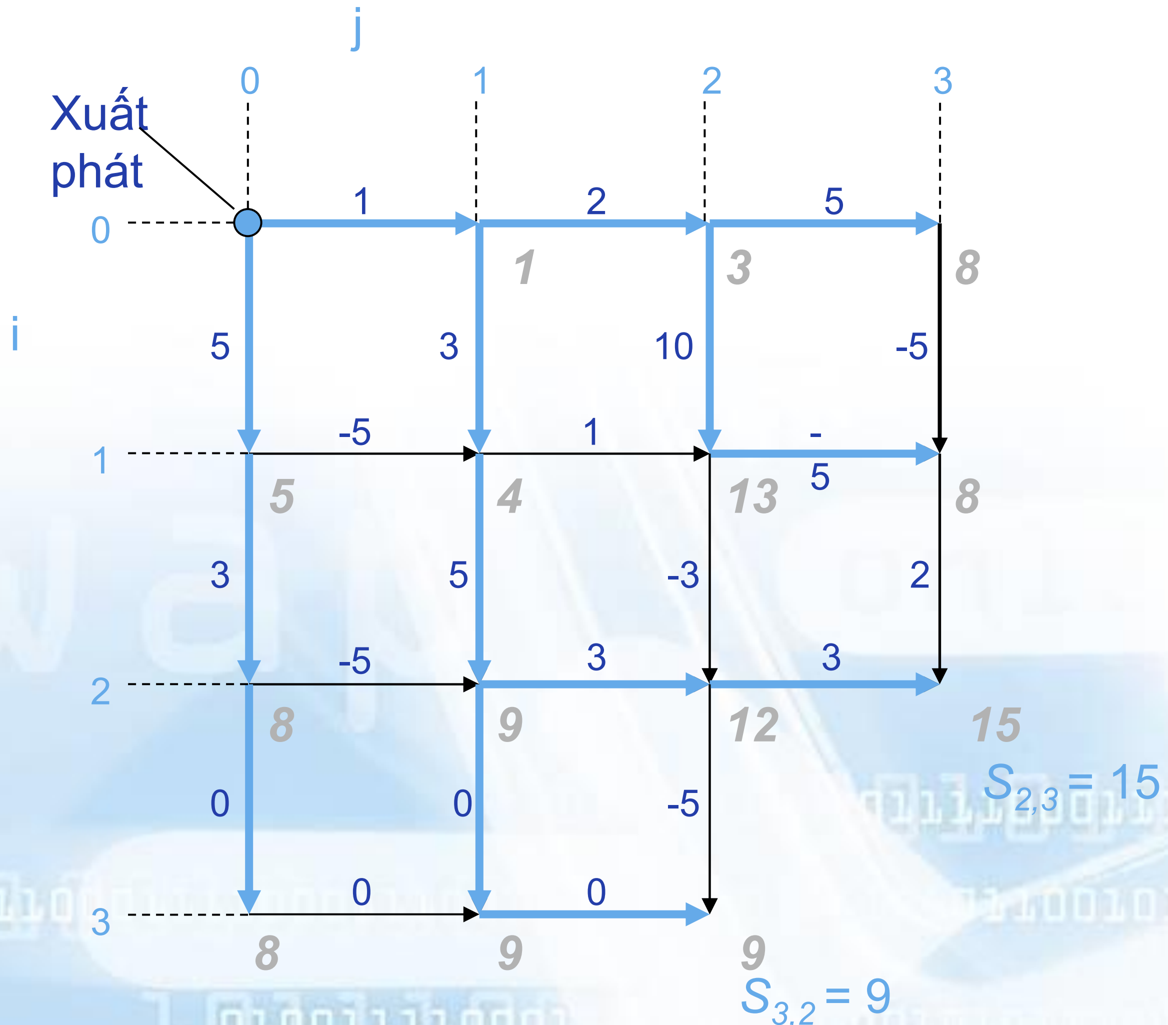
Bài toán người du lịch Manhattan



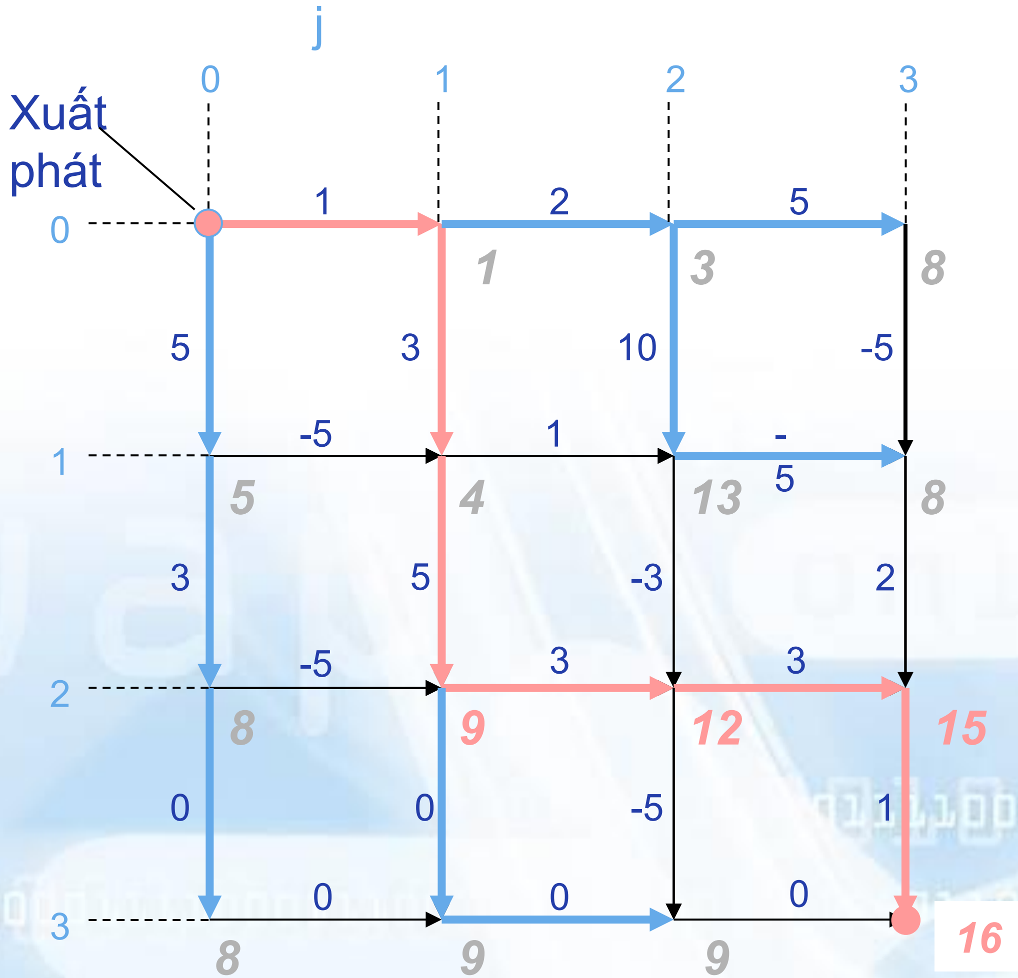
Bài toán người du lịch Manhattan



Bài toán người du lịch Manhattan



Bài toán người du lịch Manhattan



$S_{3,3} = 16$

Hoàn thành!

(Biểu diễn các lối đi)

Bài toán người du lịch Manhattan

- Rút ra giải pháp từ bảng

Thuật toán timDuong(*n,m*)

//Tìm đường đi có trọng số lớn nhất bằng giải thuật quy hoạch động

//Đầu vào: cặp số nguyên dương n, m ≥ 0, ma trận trọng số W, ma trận nút: N

//Đầu ra: giá trị Manha(*n*,m)

SoDD[0,0]=0;

for j← 1 **to** n **do**

//tính trọng số đi qua các đỉnh dòng số 0

SoDD [0, *j*] ← SoDD [0*, j-1*] + W[N[0*, j*], N[0 *,j-1*]];

//tính trọng số đi qua các đỉnh côt số 0

SoDD[j,0] ← SoDD[*j-1, 0*] + W[N[j, 0], N[j-1, 0]];

//tính trọng số đi qua các đỉnh các dòng cột còn lại

for *i*← 1 **to** n **do**

for *j* ←1 **to** m **do**

SoDD [*i* , *j*] ← max(SoDD[*i*, *j*-1] + W[N[i,j-1], N[i,j]], SoDD[*i*-1, *j*] + W[N[i-1,j], N[i,j]])

return SoDD [n, m]

Bài toán người du lịch Manhattan

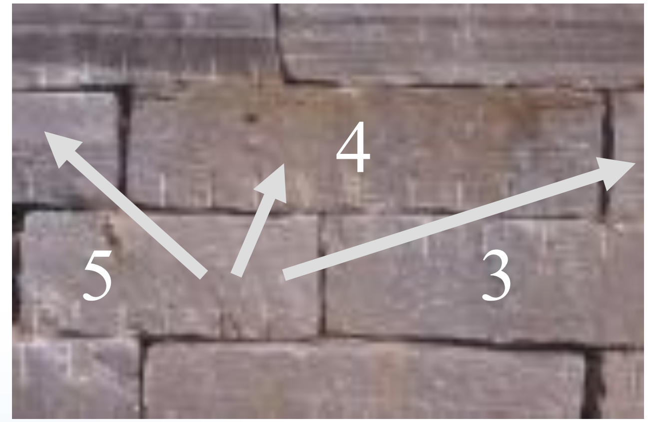
- Cài đặt bằng C/C++

Bài toán leo núi

- Có một nhà leo núi cần leo từ chân núi lên tới đỉnh theo lối an toàn nhất. Mỗi bước, nhà leo núi có thể bám vào các hốc đá ở bên trên anh ta và chỉ có thể bám vào những hốc đá gần nhất. Mỗi hốc có độ an toàn khác nhau.

Bài toán leo núi

- Ví dụ: trường hợp bên
  - Tại mỗi bước anh ta chỉ có thể tiếp cận 3 hốc đá: trên, trên phải, trên trái.
  - Có một bảng chỉ ra mức độ nguy hiểm của các hốc đá.
  - Yêu cầu tìm ra lối đi ít nguy hiểm nhất.



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

Chọn tham lam, không cho kết quả tốt

Bài toán người leo núi

- Xác định công thức đệ quy
  - $A(i,j) = C(i,j)+\min\{A(i-1,j),A(i-1,j+1)\}$ *với j=1 mép phía trái*
  - $A(i,j) = C(i,j)+\min\{A(i-1,j-1),A(i-1,j)\}$ *với j=m mép phía phải*
  - $A(i,j) = C(i,j)+\min\{A(i-1,j-1),A(i-1,j),A(i-1,j+1)\}$ *ở giữa*
  - $A(i,j) = C(i,j)$ *với i=1*
  - $A(0,j) = 0$ *đứng ở dưới đất*
  - $A(i,0)=A(i,m+1)=\infty$ không có hốc đá nên mức độ nguy hiểm rất lớn
- *Trong đó*
  - *$C(i,j)$ là mức độ nguy hiểm tại hốc đá dòng i , cột j*
  - *$A(i,j)$ là tổng chi phí nhỏ nhất của đường đi từ chân núi (dòng 1 lên tới hốc đá dòng i, cột j, $1 \le i \le n$ và $\le j \le m$,*

Bài toán người leo núi

C(i,j):

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

A(i,j):

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

Thiết lập: $A(i,0)=A(i,m+1)=\infty$, $A(0,j)=0$

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞						∞
3	∞						∞
4	∞						∞

Các giá trị ở dòng đầu là C(i,j).

Bài toán người leo núi

C(i,j):

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

A(i,j):

i \ j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7					∞
3	∞						∞
4	∞						∞

$A(2,1)=5+\min\{\infty,3,2\}=7.$

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9				∞
3	∞						∞
4	∞						∞

$A(2,1)=5+\min\{\infty,3,2\}=7$. $A(2,2)=7+\min\{3,2,5\}=9$

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7			∞
3	∞						∞
4	∞						∞

$A(2,1)=5+\min\{\infty,3,2\}=7$. $A(2,2)=7+\min\{3,2,5\}=9$
$A(2,3)=5+\min\{2,5,4\}=7$.

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7	10	5	∞
3	∞						∞
4	∞						∞

Mức độ nguy hiểm thấp nhất dòng 2 là 5.

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7	10	5	∞
3	∞	11	11	13	7	8	∞
4	∞						∞

Mức độ nguy hiểm thấp nhất dòng 3 là 7.

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7	10	5	∞
3	∞	11	11	13	7	8	∞
4	∞	13	19	16	12	15	∞

Mức độ nguy hiểm thấp nhất dòng cuối là 12.

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7	10	5	∞
3	∞	11	11	13	7	8	∞
4	∞	13	19	16	12	15	∞

Đường đi có mức độ mức độ nguy hiểm thấp nhất là 12.

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7	10	5	∞
3	∞	11	11	13	7	8	∞
4	∞	13	19	16	12	15	∞

Hốc đá cuối cùng là hốc (4,4).

Tìm ra lối đi tốt nhất bằng cách lần lại các quyết định đã làm trong quá trình tính toán A(i,j).

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7	10	5	∞
3	∞	11	11	13	7	8	∞
4	∞	13	19	16	12	15	∞

Hốc đá liền trước là hốc (3,4)
Vì min{13,7,8} là 7..

Tìm ra lối đi tốt nhất bằng cách lần lại các quyết định đã làm trong quá trình tính toán A(i,j).

VANLANG UNIVERSITY
School of Technology
Information Technology

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7	10	5	∞
3	∞	11	11	13	7	8	∞
4	∞	13	19	16	12	15	∞

Hốc đá liền trước là hốc (2,5)
Vì min{7,10,5} là 5.

Tìm ra lối đi tốt nhất bằng cách lần lại các quyết định đã làm trong quá trình tính toán A(i,j).

Bài toán người leo núi

C(i,j):

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

A(i,j):

i\j	0	1	2	3	4	5	6
0	∞	0	0	0	0	0	∞
1	∞	3	2	5	4	8	∞
2	∞	7	9	7	10	5	∞
3	∞	11	11	13	7	8	∞
4	∞	13	19	16	12	15	∞

Hốc đá liền trước là hốc (1,4)
Vì min{4,8, ∞} là 4.

Tìm ra lối đi tốt nhất bằng cách lần lại các quyết định đã làm trong quá trình tính toán A(i,j).

Chuỗi con chung không liên tiếp dài nhất

- Cho hai chuỗi S1, S2. Chuỗi C là chuỗi con chung của hai chuỗi S1, S2 nếu xóa đi một số ký tự trong chuỗi S1 thì được chuỗi C, tương tự với S2. Tìm chuỗi con chung dài nhất.
- Ví dụ: S1=abefcda và S2=ayekmcdz → C=aecd

Chuỗi con chung liên tiếp dài nhất

- Cho hai chuỗi S1, S2. Chuỗi C là chuỗi con chung của hai chuỗi S1, S2 nếu C xuất hiện trong cả S1 lẫn S2. Tìm chuỗi con chung dài nhất.
- Ví dụ: S1=abcdexy và S2=xtbayabcdez → C=abcde

Bài toán cái túi

- Cho một cái túi có khả năng chứa trọng lượng tối đa là W. Có tập S gồm n đồ vật.
- Đồ vật *i* có trọng lượng w_i và giá trị b_i (w_i, b_i, W là số nguyên)
- Bài toán: Tìm cách bỏ các đồ vật vào vừa trong túi sao cho tổng giá trị đồ vật là lớn nhất.

VANLANG UNIVERSITY
School of Technology
Information Technology

Bài toán cái túi

- Thành lập công thức đệ quy

$$B[k,w]=\begin{cases} B[k-1,w] & \text{if } w_k > w \\ \max\{B[k-1,w], B[k-1,w-w_k]+b_k\} & \text{else} \end{cases}$$

- Trong đó:
  - B[k,w] cho biết tổng giá trị lớn nhất khi xét k đồ vật đầu tiên bỏ vào túi có trong lượng w.

Bài toán cái túi

- Lập bảng giá trị

VANLANG UNIVERSITY
School of Technology
Information Technology

Bài toán cái túi

- Thuật toán

```
CaiTui(w,B,n, b)
    for w = 0 to W    //không chọn đò vật nào
        B[0,w] = 0 //thì giá trị luôn là 0 với mọi túi chứa trọng lượng w
    for i = 0 to n // có i đồ vật mà túi không có khả năng chứa
        B[i,0] = 0  // thì giá trị sử dụng là 0
        for w = 0 to W
            if wi <= w // xét đồ vật có thể bỏ vào túi với khả năng w
            //nếu chọn đồ vật thứ i cho giá trị sử dụng lớn hơn
                if bi + B[i-1,w-wi] > B[i-1,w]
                    B[i,w] = bi + B[i-1,w- wi] //thì chọn đồ vật i
                else
                    B[i,w] = B[i-1,w] //ngoài ra không chọn
            else B[i,w] = B[i-1,w]  // wi > w : túi không chứa nổi
```

Xin cám ơn