Chương 4. ĐƯỜNG ĐI NGẮN NHẤT

Thuật toán Floyd-Warshall

ThS. Nguyễn Chí Hiếu

2019

NỘI DUNG

1. Các khái niệm cơ bản
2. Thuật toán Floyd-Warshall
3. Ứng dụng của thuật toán Floyd-Warshall

Các khái niệm cơ bản

Chiến lược **Quy hoạch động** *(Dynamic programming)*

- Do Richard Bellman đề xuất nhằm giảm thời gian thực hiện các thuật toán có sử dụng đệ quy và cấu trúc con tối ưu.
- Hướng tiếp cận
  - Trên-xuống *(top-down)*
  - Dưới-lên *(bottom-up)*: bắt đầu từ các bài toán con nhỏ hơn, giải quyết đến bài toán ban đầu. Sử dụng bảng để lưu trữ kết quả của các bài toán con.

Các khái niệm cơ bản

Ví dụ 1

Tính phần tử thứ n của dãy Fibonacci theo công thức:

$$f(n) = \begin{cases} n & , n = 0, 1 \\ f(n-1) + f(n-2) & , n > 1 \end{cases}$$

Các khái niệm cơ bản

- Sử dụng kỹ thuật đệ quy.

Thuật toán 1: **Fib(n)**

- Đầu vào: n phần tử.

- Đầu ra: giá trị phần tử thứ n của dãy Fibonacci.

```
if n = 0 or n = 1
    return n
return Fib(n - 1) + Fib(n - 2)
```

Các khái niệm cơ bản

- Sử dụng kỹ thuật quy hoạch động.

Thuật toán 2: **Fib1(n)**

- Đầu vào: n phần tử.
- Đầu ra: giá trị phần tử thứ n của dãy Fibonacci.

```
f[0] ← 0
f[1] ← 1
for i ← 2 to n
    f[i] ← f[i - 1] + f[i - 2]
return f[n]
```

Các khái niệm cơ bản

- Sử dụng kỹ thuật quy hoạch động.

Thuật toán 3: **Fib2(n)**

- Đầu vào: n phần tử.

- Đầu ra: giá trị phần tử thứ n của dãy Fibonacci.

```
preFib ← 0
curFib ← 1
for i ← 2 to n
    newFib = preFib + curFib
    preFib = curFib
    curFib = newFib
return curFib
```

Thuật toán Floyd-Warshall

Bài toán tìm đường đi ngắn nhất giữa các cặp đỉnh trong đồ thị

- Sử dụng n lần các thuật toán Dijkstra hay Bellman-Ford sẽ không hiệu quả.

Thuật toán Floyd-Warshall [1]

Ý tưởng

Cho đồ thị $G = (V, E)$. Xét hai đỉnh $i, j \in V$, tìm đường đi ngắn nhất giữa hai đỉnh i và j.

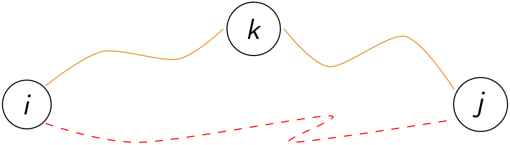
- Giả sử $k \in V$ là đỉnh thuộc đường đi ngắn nhất từ $i \rightarrow j$. Khi đó, đường đi ngắn nhất giữa hai đỉnh i, j là $d[i, j]$ được tính bằng *đường đi ngắn nhất giữa hai cặp đỉnh* $\{d[i, k], d[k, j]\}$ như sau:

$$\boxed{d[i, j] = min\{d[i, j], d[i, k] + d[k, j]\}.} \tag{1}$$

[1] Robert W. Floyd (1936 – 2001), Stephen Warshall (1935 – 2006)

Thuật toán Floyd-Warshall

Ví dụ 2



Hình 1: Đường đi ngắn nhất từ $i \rightsquigarrow k \rightsquigarrow j$.

Thuật toán Floyd-Warshall

Thuật toán 4: **Floyd-Warshall(G, start, end)**

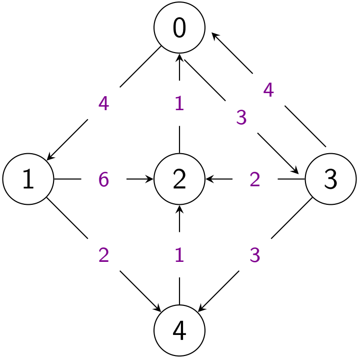
- Đầu vào: đồ thị G, đỉnh start, đỉnh end.
- Đầu ra: đường đi ngắn nhất từ đỉnh start → end.

```
for each i ∈ V
    for each j ∈ V
        p[i][j] ← NO_PATH
        if i = j
            d[i][j] ← 0
        else if edge(i,j) ∈ E
            d[i][j] ← w[i][j]
        else
            d[i][j] ← ∞
            p[i][j] ← i
for each k ∈ V
    for each i ∈ V
        for each j ∈ V
            d[i][j] ← min{d[i][j], d[i][k] + d[k][j]}
            p[i][j] ← p[k][j]
```

Thuật toán Floyd-Warshall

Ví dụ 3

Cho đồ thị G, tìm đường đi ngắn nhất từ đỉnh 0 đến đỉnh 4.



Hình 2: Đồ thị có hướng G.

Thuật toán Floyd-Warshall

- Bước 1: Khởi tạo giá trị cho ma trận khoảng cách giữa các cặp đỉnh bất kỳ.

$$D = \begin{array}{|ccccc|} \hline 0 & 4 & \infty & 3 & \infty \\ \infty & 0 & 6 & \infty & 2 \\ 1 & \infty & 0 & \infty & \infty \\ 4 & \infty & 2 & 0 & 3 \\ \infty & \infty & 1 & \infty & 0 \\ \hline \end{array}$$

- Bước 2: Lặp k lần để tính ma trận D_k. Trong đó D_k là ma trận tương ứng khoảng cách ngắn nhất giữa 2 cặp đỉnh i và j với $k = [0, 4]$ là đỉnh trung gian.

Thuật toán Floyd-Warshall

- $D_0[2, 1] = \min\{D_0[2, 1], D_0[2, 0] + D_0[0, 1]\} = 5$

$$D_0 = \begin{bmatrix} 0 & 4 & \infty & 3 & \infty \\ \infty & 0 & 6 & \infty & 2 \\ 1 & \mathbf{5} & 0 & \mathbf{4} & \infty \\ 4 & \mathbf{8} & 2 & 0 & 3 \\ \infty & \infty & 1 & \infty & 0 \end{bmatrix} \quad P_0 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & \mathbf{0} & -1 & \mathbf{0} & -1 \\ -1 & \mathbf{0} & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} 0 & 4 & \mathbf{10} & 3 & \mathbf{6} \\ \infty & 0 & 6 & \infty & 2 \\ 1 & 5 & 0 & 4 & \mathbf{7} \\ 4 & 8 & 2 & 0 & 3 \\ \infty & \infty & 1 & \infty & 0 \end{bmatrix} \quad P_1 = \begin{bmatrix} -1 & -1 & \mathbf{1} & -1 & \mathbf{1} \\ -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & -1 & 0 & \mathbf{1} \\ -1 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Thuật toán Floyd-Warshall

$$D_2 = \begin{bmatrix} 0 & 4 & 10 & 3 & 6 \\ \mathbf{7} & 0 & 6 & \mathbf{10} & 2 \\ 1 & 5 & 0 & 4 & 7 \\ \mathbf{3} & \mathbf{7} & 2 & 0 & 3 \\ \mathbf{2} & \mathbf{6} & 1 & \mathbf{5} & 0 \end{bmatrix} \qquad P_2 = \begin{bmatrix} -1 & -1 & 1 & -1 & 1 \\ \mathbf{2} & -1 & -1 & \mathbf{2} & -1 \\ -1 & 0 & -1 & 0 & 1 \\ \mathbf{2} & \mathbf{2} & -1 & -1 & -1 \\ \mathbf{2} & \mathbf{2} & -1 & \mathbf{2} & -1 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} 0 & 4 & \mathbf{5} & 3 & 6 \\ 7 & 0 & 6 & 10 & 2 \\ 1 & 5 & 0 & 4 & 7 \\ 3 & 7 & 2 & 0 & 3 \\ 2 & 6 & 1 & 5 & 0 \end{bmatrix} \qquad P_3 = \begin{bmatrix} -1 & -1 & \mathbf{3} & -1 & 1 \\ 2 & -1 & -1 & 2 & -1 \\ -1 & 0 & -1 & 0 & 1 \\ 2 & 2 & -1 & -1 & -1 \\ 2 & 2 & -1 & 2 & -1 \end{bmatrix}$$

Thuật toán Floyd-Warshall

$$D_4 = \begin{array}{|cccc|c|} 0 & 4 & 5 & 3 & 6 \\ \mathbf{4} & 0 & \mathbf{3} & \mathbf{7} & 2 \\ 1 & 5 & 0 & 4 & 7 \\ 3 & 7 & 2 & 0 & 3 \\ \hline 2 & 6 & 1 & 5 & 0 \end{array} \qquad P_4 = \begin{array}{|rrrrr|} -1 & -1 & 3 & -1 & 1 \\ \mathbf{4} & -1 & \mathbf{4} & \mathbf{4} & -1 \\ -1 & 0 & -1 & 0 & 1 \\ 2 & 2 & -1 & -1 & -1 \\ 2 & 2 & -1 & 2 & -1 \end{array}$$

- Đường đi ngắn nhất $0 \to 4$: $(0, 1, 4)$.
- Chiều dài đường đi ngắn nhất: 6.

Độ phức tạp của thuật toán

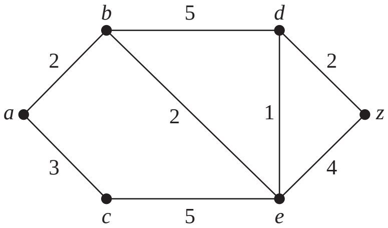
- Thuật toán chứa 3 vòng lặp nên độ phức tạp là $\mathcal{O}\left(n^3\right)$.

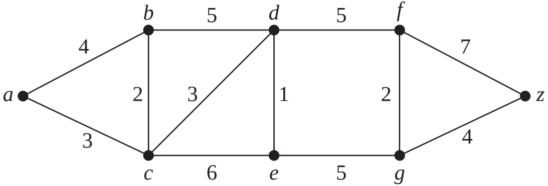
Ứng dụng của thuật toán Floyd-Warshall

- Tìm đường đi ngắn nhất giữa các đỉnh trong đồ thị.
- Phát hiện chu trình âm trong đồ thị: sau khi thực hiện thuật toán, nếu $D[i][i] < 0$ thì đồ thị có chu trình âm bắt đầu và kết thúc tại đỉnh i.

Bài tập

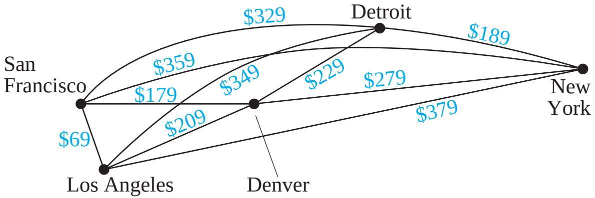
1. Tìm đường đi ngắn nhất giữa *a* và *z* trong các đồ thị có trọng số sau đây:

   a) 

   b) 

Bài tập

2. Tìm đường bay có chi phí nhỏ nhất giữa các thành phố.



Tài liệu tham khảo

- ADRIAN BONDY, U.S.R. MURTY, *Graph Theory*, Springer, 2008.
- KENNETH H. ROSEN, *Discrete Mathematics and its Applications, 7th Edidion*, McGraw-Hill, 2011.
- NGUYỄN ĐỨC NGHĨA, NGUYỄN TÔ THÀNH, *Toán rời rạc*, NXB Đại học Quốc gia Hà Nội, 2003.
- NGUYỄN CAM, CHU ĐỨC KHÁNH, *Lý thuyết đồ thị*, NXB Đại học Quốc gia Tp Hồ Chí Minh, 2008.
- REINHARD DIESTEL, *Graph Theory*, Springer, 2005.