Graph Algorithm IV

Instructor: Shizhe Zhou

Course Code:00125401

All-pair shortest path

- 对无向或有向图，求每对节点间的最短路径。

- 对途经的最大顶点index做逐步的松弛

- Dynamic Programming :

简单的数据结构！：可直接使用邻接矩阵存储或者边表存储(our code).

最短路径与矩阵乘法

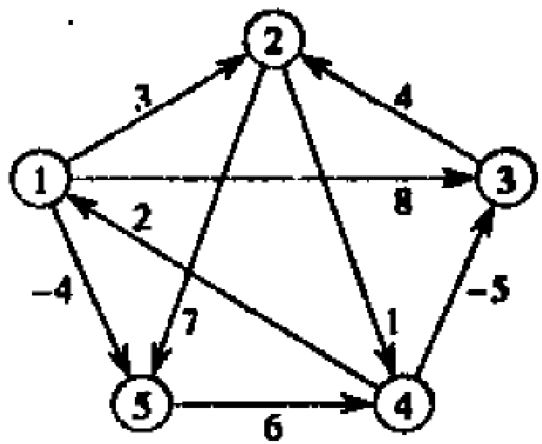
$$d_{ij}^{(m)} = \min\left(d_{ij}^{(m-1)}, \min_{1\le k\le n}\left\{d_{ik}^{(m-1)} + w_{kj}\right\}\right)$$

$$= \min_{1\le k\le n}\left\{d_{ik}^{(m-1)} + w_{kj}\right\}$$

$d^{(m-1)}$	$\rightarrow a$
w	$\rightarrow b$
$d^{(m)}$	$\rightarrow c$
min	$\rightarrow +$
$+$	$\rightarrow \cdot$

```
EXTEND-SHORTEST-PATHS(D,W)
n←rows[D]
设 D′=(d′ij)是一个 n×n 矩阵
for i←1 to n
    do for j←1 to n
          do d′ij←∞
             for k←1 to n
                 do d′ij←min(d′ij,d[ik+wkj)
return D′
```

```
MATRIX-MULTIPLY(A,B)
n←rows[A]
设 C 为一 n×n 矩阵
for i←1 to n
    do for j←1 to n
          do cij←0
             for k←1 to n
                 do cij←cij+aik · bkj
return C
```



$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad D^{(4)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

算法运行时间的改进

目标并不是计算出全部的 $D^{(m)}$ 矩阵

我们所感兴趣的仅仅是矩阵 $D^{(n-1)}$

通过计算下列矩阵序列我们可以仅计算 $\lceil \lg(n-1) \rceil$ 个矩阵积就得

$$
\begin{aligned}
D^{(1)} &= W \\
D^{(2)} &= W^{2} = W \cdot W \\
D^{(4)} &= W^{4} = W^{2} \cdot W^{2} \\
D^{(8)} &= W^{8} = W^{4} \cdot W^{4} \\
&\cdots \\
D^{(2^{\lceil \lg(n-1) \rceil})} &= W^{2^{\lceil \lg(n-1) \rceil}} = W^{2^{\lceil \lg(n-1) \rceil - 1}} \cdot W^{2^{\lceil \lg(n-1) \rceil - 1}}
\end{aligned}
$$

Complexity,小常数

$\Theta(n^3 \lg n)$

引理

- 最短路径的子路径也是最短路径
- 证明：可反证法直接证明.

Floyd-Warshall

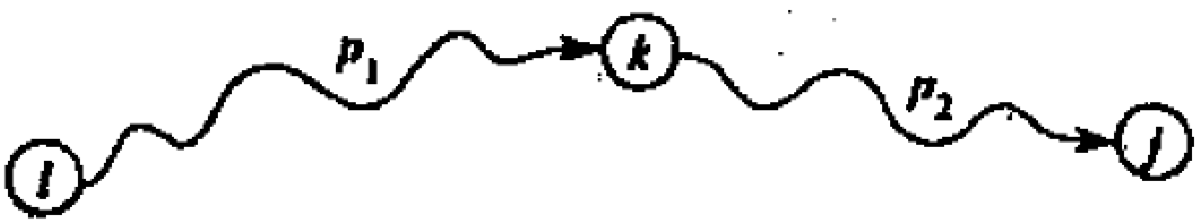
解决每对结点间最短路径问题的一种递归方案

- 对中间结点的指标数分类，递增的递归求最短路径.

归纳: 已知任意点对的中间节点属于(0,..,k-1)的最短路径长度. k-1 path

归纳基础: **当 $k=0$ 时，从结点 i 到结点 j 的路径中根本不存在中间结点**

路径 p 分解为 $i \overset{p_1}{\rightsquigarrow} k \overset{p_2}{\rightsquigarrow} j$.



If k不是p的中间节点,则p仍是k-1 path, 否则，p可作如左图之分解

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{若} k=0 \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{若} k \geq 1 \end{cases}$$

自底向上计算最短路径的权

Floyd − Warshall(W)

n ← rows[W]

$D^{(0)} \leftarrow W$

for k ← 1 to n

  do for i ← 1 to n

    do for j ← 1 to n

$$d_{ij}^{(k)} \leftarrow \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

return $D^{(n)}$

Complexity,小常数

$\Theta(n^3)$

优于

$\Theta(n^3 \lg n)$

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(0)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & \text{NIL} & 4 & \text{NIL} & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

initialization

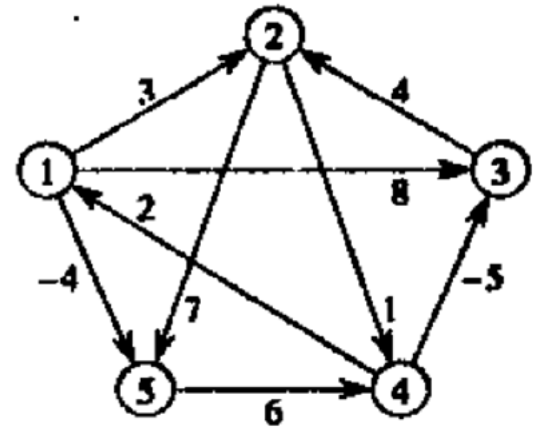
$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(1)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & 1 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(2)} = \begin{pmatrix} \text{NIL} & 1 & 1 & 2 & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & 2 & 2 \\ 4 & 1 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(3)} = \begin{pmatrix} \text{NIL} & 1 & 1 & 2 & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & 2 & 2 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad \Pi^{(4)} = \begin{pmatrix} \text{NIL} & 1 & 4 & 2 & 1 \\ 4 & \text{NIL} & 4 & 2 & 1 \\ 4 & 3 & \text{NIL} & 2 & 1 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ 4 & 3 & 4 & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad \Pi^{(5)} = \begin{pmatrix} \text{NIL} & 3 & 4 & 5 & 1 \\ 4 & \text{NIL} & 4 & 2 & 1 \\ 4 & 3 & \text{NIL} & 2 & 1 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ 4 & 3 & 4 & 5 & \text{NIL} \end{pmatrix}$$



- 算法all-pair shortest path method(floyd-warshall)

Direct related to transitive closure

算法 *All_Pairs_Shortest_Paths (Weight)*

输入： *Weight*（表示一个加权图的 $n \times n$ 邻接矩阵）

{*Weight*[x, y]是边(x, y)的权重，如果该边不存在，则其值为∞，另外对于所有 x，*Weight*[x, x]的值为 0}

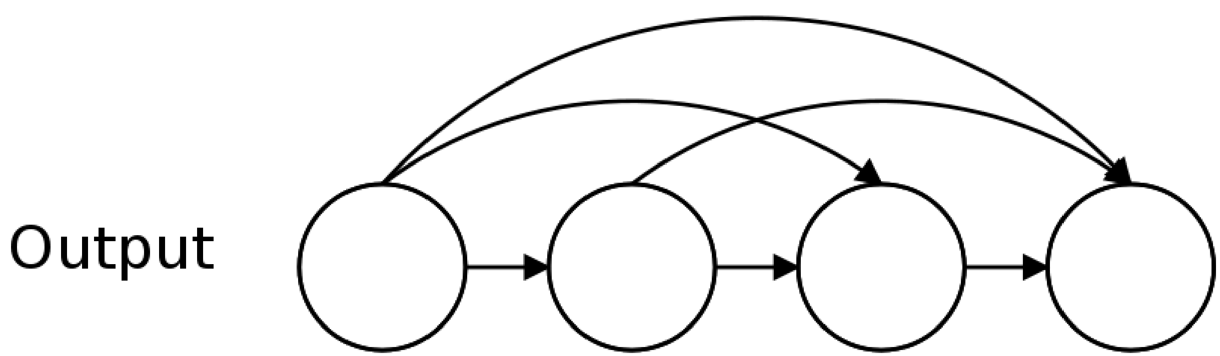
输出： 最终，矩阵 *Weight* 包含所有节点对的最短路径的值

```
begin
    for m := 1 to n do {归纳序列}
        for x := 1 to n do
            for y := 1 to n do
                if Weight[x, m] + Weight[m, y] < Weight[x, y] then
                    Weight[x, y] := Weight[x, m] + Weight[m, y]
end
```

图 7.22　全部最短路径算法

传递闭包

- 有向C(V,F)是有向G=(V,E)的传递闭包: (v,w)属于F，当且仅当v到w在G中存在一条路径.



- 算法改进自all-pair shortest path method

min 和 + 用相应的逻辑运算 $\vee$ 和 $\wedge$ 来代替

Modify this to do boolean operation!

算法 ***All_Pairs_Shortest_Paths (Weight)***

输入： *Weight*（表示一个加权图的 $n \times n$ 邻接矩阵）

{*Weight*[x, y]是边(x, y)的权重，如果该边不存在，则其值为∞，另外对于所有 x，*Weight*[x, x]的值为 0}

输出： 最终，矩阵 *Weight* 包含所有节点对的最短路径的值

```
begin
    for m := 1 to n do {归纳序列}
        for x := 1 to n do
            for y := 1 to n do
                if Weight[x, m] + Weight[m, y] < Weight[x, y] then
                    Weight[x, y] := Weight[x, m] + Weight[m, y]
end
```

图 7.22 全部最短路径算法

算法 ***Transitive_Closure (A)***

输入： *A*（表示一个有向图的 $n \times n$ 邻接矩阵）

{若图包含边(*x*, *y*)，则 *A*[*x*, *y*]为真，否则为假，对于所有 *x*，*A*[*x*, *x*]为真}

输出： 最终，矩阵 *A* 的值为图的传递闭包

```
begin
    for m := 1 to n do {归纳序列}
        for x := 1 to n do
            for y := 1 to n do
                if A[x, m] and A[m, y] then A[x, y] := true
                {这一步将在后续算法中得以改进}
end
```

Complexity,小常数

$\Theta(n^3)$

2 ACCESS, SLOW

算法 ***Improved_Transitive_Closure (A)***

输入： *A*（表示一个有向图的 $n \times n$ 邻接矩阵）

{若图包含边(*x*, *y*)，则 *A*[*x*, *y*]为真，否则为假，对于所有 *x*，*A*[*x*, *x*]为真}

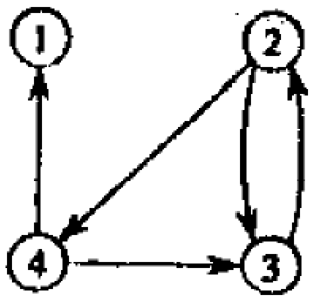
输出： 最终，矩阵 *A* 的值为图 *G* 的传递闭包

```
begin
    for m := 1 to n do {归纳序列}
        for x := 1 to n do
            if A[x, m] then
                for y := 1 to n do
                    if A[m, y] then A[x, y] := true
end
```

EARLY retreat

Examples on transitive closure

- A directed graph



$$T^{(0)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad T^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad T^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

$$T^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad T^{(4)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$