# Algorithms
# Chapter 25
# All-Pairs Shortest Paths

Associate Professor: Ching-Chi Lin
林清池 副教授
chingchi.lin@gmail.com

Department of Computer Science and Engineering
National Taiwan Ocean University

# Outline

- **Shortest paths and matrix multiplication**
- The Floyd-Warshall algorithm
- Johnson’s algorithm for sparse graphs

# Overview$_{1/2}$

- **Input:** A weighted directed graph $G = (V, E)$.
- **Output:** An $n \times n$ matrix of shortest-path distances $\delta(u,v)$.
- Could run BELLMAN-FORD once from each vertex:
  - $O(n^2 m)$ — which is $O(n^4)$ if the graph is **dense** ($E = \Theta(n^2)$).
- If no negative-weight edges, could run Dijkstra's algorithm algorithm once from each vertex:
  - $O(nm \lg n)$ with binary heap — $O(n^3 \lg n)$ if dense.
  - $O(n^2 \lg n + nm)$ with Fibonacci heap — $O(n^3)$ if dense.
- We'll see how to do in $O(n^3)$ in all cases, with no fancy data structure.

# Overview$_{2/2}$

- **Input:** The adjacency matrix $W$ of a weighted directed graph $G = (V, E)$, where

$$w_{ij} = \begin{cases} 0 & \text{if } i = j, \\ \text{the weight of edge } (i, j) & \text{if } i \neq j \text{ and } (i, j) \in E, \\ \infty & \text{if } i \neq j \text{ and } (i, j) \notin E. \end{cases}$$

  - Negative weights – "allow". Negative-weight cycles – "no".

- **Output:** A matrix $D = (d_{ij})$, where $d_{ij} = \delta(i, j)$.

# Shortest paths and matrix multiplication

- A dynamic programming approach
- **Optimal substructure:** subpaths of shortest paths are shortest paths.
- **Recursive solution:** Let $l_{ij}^{(m)}$ = weight of shortest path from *i* to *j* that contains at most *m* edges.
  - $m = 0$, there is a shortest path from *i* to *j* with no edges if and only if $i = j$ .

$$l_{ij}^{(0)} = \begin{cases} 0 & \text{if } i = j, \\ \infty & \text{if } i \neq j. \end{cases}$$

  - $m \geq 1$, $l_{ij}^{(m)} = \min\{l_{ij}^{(m-1)}, \min_{1 \leq k \leq n}\{l_{ik}^{(m-1)} + w_{kj}\}\}$

$$= \min_{1 \leq k \leq n}\{l_{ik}^{(m-1)} + w_{kj}\} \quad (\text{since } w_{jj} = 0).$$

  - $m = 1$, we have $l_{ij}^{(1)} = w_{ij}$.

# Compute a solution bottom-up$_{1/2}$

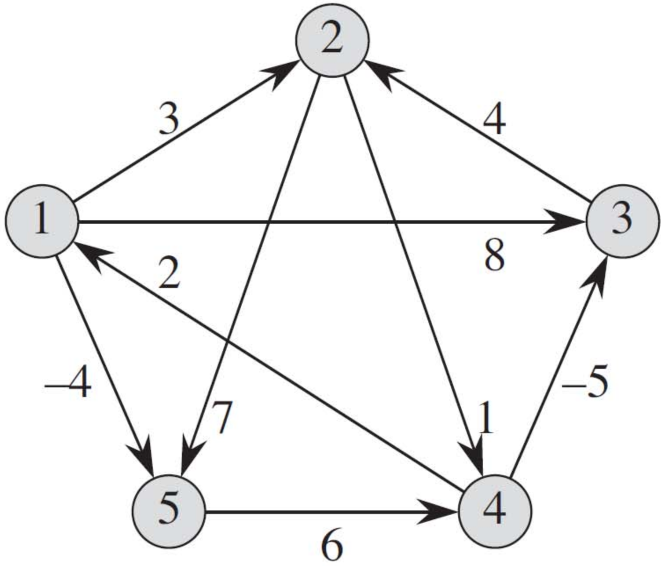
- $d_{ij} = \delta(i, j) = l_{ij}^{(n-1)} = l_{ij}^{(n)} = l_{ij}^{(n+1)} = \ldots$
- Compute $L^{(1)}, L^{(2)}, \ldots, L^{(n-1)}$, where $L^{(1)} = W$.

All simple shortest paths contain at most $n - 1$ edges.

- Go from $L^{(m-1)}$ to $L^{(m)}$.

```
EXTEND-SHORTEST-PATHS(L, W)
    n ← rows[L]
    let L' = (l'_ij) be an n × n matrix
    for i ← 1 to n
        for j ← 1 to n
            l'_ij ← ∞
            for k ← 1 to n
                l'_ij ← min{l'_ij, l_ik + w_kj}
    return L'
```

- $L$ for $L^{(m-1)}$ and $L'$ for $L^{(m)}$.
- Time: $\Theta(n^3)$.




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

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

# Compute a solution bottom-up$_{2/2}$

- **Observation:** Extend is like matrix multiplication: $C = A \cdot B$.
  - $L \rightarrow A$, $W \rightarrow B$, $L' \rightarrow C$
  - $\min \rightarrow +$, $+ \rightarrow *$, $\infty \rightarrow 0$

Square-Matrix-Multiply$(A, B)$

$n \leftarrow \text{rows}[A]$<br>
let $C$ be an $n \times n$ matrix<br>
**for** $i \leftarrow 1$ to $n$<br>
    **for** $j \leftarrow 1$ to $n$<br>
        $c_{ij} \leftarrow 0$<br>
        **for** $k \leftarrow 1$ to $n$<br>
            $c_{ij} \leftarrow c_{ij} + a_{ik} * b_{kj}$<br>
return $C$

Extend-Shortest-Paths$(L, W)$

$n \leftarrow \text{rows}[L]$<br>
let $L' = (l'_{ij})$ be an $n \times n$ matrix<br>
**for** $i \leftarrow 1$ to $n$<br>
    **for** $j \leftarrow 1$ to $n$<br>
        $l'_{ij} \leftarrow \infty$<br>
        **for** $k \leftarrow 1$ to $n$<br>
            $l'_{ij} \leftarrow \min\{l'_{ij}, l_{ik} + w_{kj}\}$<br>
return $L'$

- $L^{(1)} = L^{(0)} \cdot W = W$, $L^{(2)} = L^{(1)} \cdot W = W^{(2)}$, $L^{(3)} = L^{(2)} \cdot W = W^{(3)}$ $L^{(n-1)} = L^{(n-2)} \cdot W = W^{(n-1)}$.
- Call Extend $n-1$ times, $D = W^{(n-1)}$ can be computed in $\Theta(n^4)$ time.

# Improving the running time

- **Goal:** to compute $W^{(n-1)}$.
  - Don't need to compute **all** the intermediate $W^{(1)}, W^{(2)}, \ldots, W^{(n-2)}$.
  - Could compute $W^2 = W \cdot W$, $W^{(4)} = W^{(2)} \cdot W^{(2)}$, $W^{(8)} = W^{(4)} \cdot W^{(4)}, \ldots, W^{2^{\lceil \lg(n-1) \rceil}} = W^{2^{\lceil \lg(n-1) \rceil - 1}} \cdot W^{2^{\lceil \lg(n-1) \rceil - 1}}$
  - Only $\lceil \lg(n-1) \rceil$ matrix products is computed.
  - Since $2^{\lceil \lg(n-1) \rceil} \geq n-1$, the final product is equal to $W^{(n-1)}$.

FASTER-ALL-PAIRS-SHORTEST-PATHS($W$) **Time :** $\Theta(n^3 \lg n)$

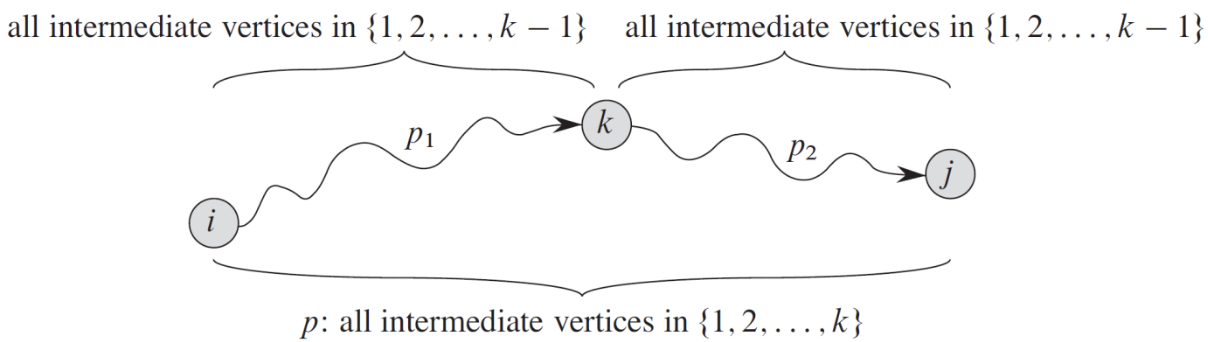
```
n ← rows[W]            ⎫
L^(1) ← W              ⎬ O(1)
m ← 1                  ⎭
while m < n − 1                                        ⎫
    do L^(2m) = EXTEND-SHORTEST-PATHS(L^(m), L^(m))    ⎬ ⌈lg(n − 1)⌉ · Θ(n^3)
        m ← 2m                                         ⎭
return L^(m)
```

# Outline

- Shortest paths and matrix multiplication
- **The Floyd-Warshall algorithm**
- Johnson’s algorithm for sparse graphs

# Floyd-Warshall algorithm

- A different dynamic-programming approach
- Let $d_{ij}^{(k)}$ be the weight of a shortest path from *i* to *j* with all intermediate vertices in {1, 2, . . . , *k*}.
- Consider a shortest path *p* from *i* to *j* with all intermediate vertices in {1, 2, . . . , *k*}:
  - If *k* is not an intermediate vertex, then all intermediate vertices of *p* are in {1, 2, . . . , *k*−1}.
  - If *k* is an intermediate vertex, then



# Recursive formulation

- A recursive solution:

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

- $d_{ij}^{(0)} = w_{ij}$, because have no intermediate vertices.
  - Such a path has ≤ 1 edge.
- Goal: $D^{(n)} = (d_{ij}^{(n)})$
  - Because for any path, all intermediate vertices are in the set {1, 2,…, *n*}.

# Compute bottom up

- Compute the values $d_{ij}^{(k)}$ in order of increasing values of $k$.

FLOYD-WARSHALL($W$)

```
n ← rows[W]
D^(0) ← W
for k ← 1 to n
    for i ← 1 to n
        for j ← 1 to n
            d_ij^(k) = min(d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1))
return D(n)
```

Lines 1–2: $O(1)$

Lines 3–6: $\Theta(n^3)$

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

- Time: $\Theta(n^3)$.

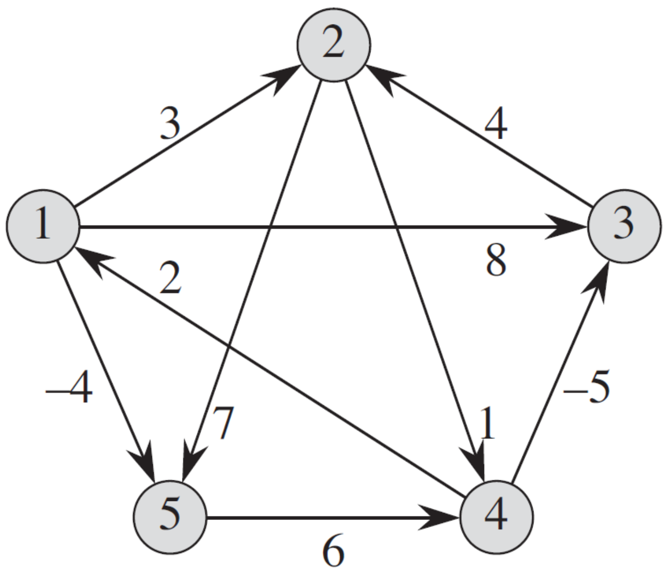
# Constructing a shortest path

- $\pi_{ij}^{(k)}$ is the predecessor of vertex $j$ on a shortest path from vertex $j$ with all intermediate vertices in the set $\{1,2,\cdots,k\}$.

$$\pi_{ij}^{(0)} = \begin{cases} \text{NIL} & \text{if } i = j, \text{or } w_{ij} = \infty, \\ i & \text{if } i \neq j, \text{and } w_{ij} < \infty. \end{cases}$$

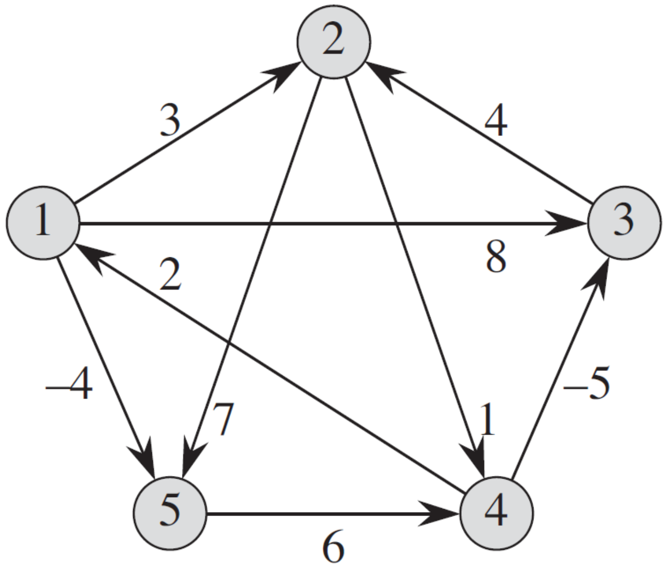
- For $k \geq 1$, we have

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}. \end{cases}$$



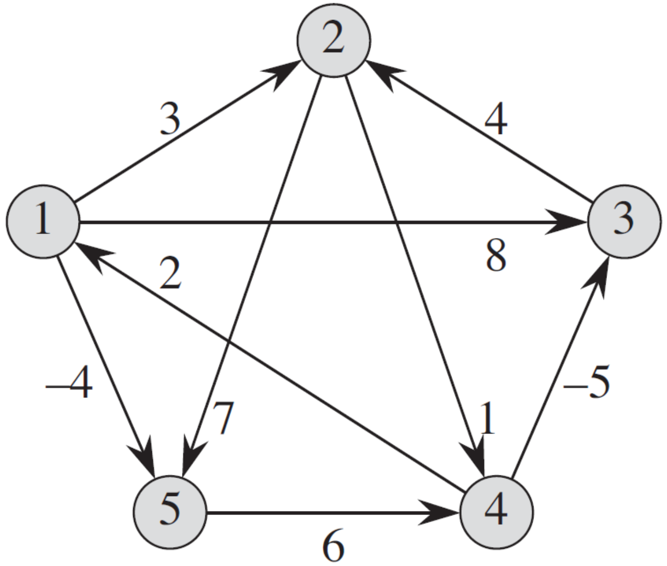

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(0)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL}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

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




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

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




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

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

# Transitive closure$_{1/2}$

- The transitive closure of $G$ is defined as the graph $G^* = (V, E^*)$, where

  $E^* = \{(i, j) : \text{there is a path from vertex } i \text{ to vertex } j \text{ in } G\}$.

- We give two methods to compute the transitive closure of a graph in the following, both in $\Theta(n^3)$ time.

- **Method 1:**
  - Assign a weight of 1 to each edge, then run FLOYD-WARSHALL.
  - If there is a path from vertex $i$ to vertex $j$, we get $d_{ij} < n$.
  - Otherwise, we get $d_{ij} = \infty$.

# Transitive closure$_{2/2}$

- **Method 2:**
  - Save time and space in practice.
  - Substitute other values and operators in FLOYD-WARSHALL.
    - min $\rightarrow$ $\vee$ (OR)
    - $+$ $\rightarrow$ $\wedge$ (AND)
  - $$t_{ij}^{(k)} = \begin{cases} 1 & \text{if there exists a path from } i \text{ to } j \text{ with all intermediate} \\ & \text{vertices in } \{1, 2, \ldots, k\}, \\ 0 & \text{otherwise.} \end{cases}$$
  - $$t_{ji}^{(0)} = \begin{cases} 0 & \text{if } i \neq j \text{ and } (i, j) \notin E, \\ 1 & \text{if } i = j \text{ or } (i, j) \in E. \end{cases}$$
  - $$t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{ki}^{(k-1)}).$$

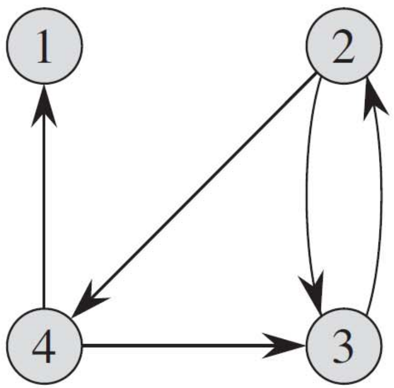
# Compute bottom up

- Compute the values $t_{ij}^{(k)}$ in order of increasing values of $k$.

TRANSITIVE-CLOSURE($G$)

$n \leftarrow |V[G]|$ } $O(1)$<br>
**for** $i \leftarrow 1$ **to** $n$<br>
    **for** $j \leftarrow 1$ **to** $n$<br>
        **if** $i = j$ or $(i, j) \in E[G]$<br>
            $t_{ij}^{(0)} \leftarrow 1$<br>
        **else** $t_{ij}^{(0)} \leftarrow 0$<br>
} $\Theta(n^2)$<br>
**for** $k \leftarrow 1$ **to** $n$<br>
    **for** $i \leftarrow 1$ **to** $n$<br>
        **for** $j \leftarrow 1$ **to** $n$<br>
            $t_{ij}^{(k)} \leftarrow t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)})$<br>
} $\Theta(n^3)$<br>
**return** $T(n)$

- Time: $\Theta(n^3)$.
- Only 1 bit is required for each $t_{ij}^{(k)}$.
- $G^*$ can be used to determine the strongly connected components of $G$.




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

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