All pairs shortest paths.

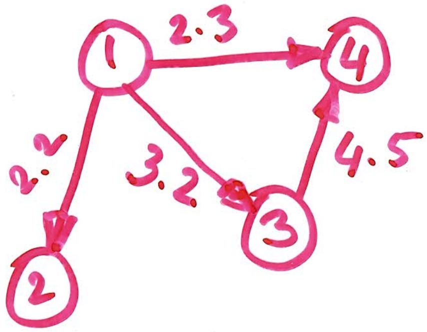
$O(n^2 \log n + mn)$ time by Dijkstra's algorithm (only with non-negative weights)

$O(mn)$ by Bellman-Ford (with negative wts. no negative cycle)

$O(n^3 \log n)$ by repeated matrix multiplication

$O(n^3)$ by Floyd-Warshall's algorithm.

Input: Adjacency matrix W.



	1	2	3	4
1	0	2.2	3.2	2.3
2	∞	0	∞	∞
3	∞	∞	0	4.5
4	∞	∞	∞	0

Output: Matrix D where

d_{ij} = wt. of the shortest path from i to j.

Shortest path with repeated matrix multiplication:

Let $d_{ij}^{(m)}$ be the wt. of the s.p. from i to j that contains at most m edges.

$$d_{ij}^{(1)} = \begin{cases} 0 & \text{if } i = j \\ w_{ij} & \text{if } i \neq j \end{cases}$$

$$d_{ij}^{(m)} = \min \begin{cases} d_{ij}^{(m-1)}, \\ \min\limits_{1 \le k \le n} \{ d_{ik}^{(m-1)} + w_{kj} \} \end{cases} \quad \text{for } m \geq 2$$

$$\delta(i,j) = d_{ij}^{(n-1)} = d_{ij}^{(n)} = \cdots$$

Extend-shortest-paths $(D^{(m)}, W)$

Let $D^{(m+1)} = (d_{ij}^{(m+1)})$ be the $n \times n$ matrix

$O(n^3)$

```
for i := 1 to n
    for j := 1 to n
        d_ij^(m+1) := ∞
        for k := 1 to n
            d_ij^(m+1) := min(d_ij^(m+1), d_ik^(m) + w_kj)
        endfor
    endfor
endfor
return D^(m+1)
```

For matrix multiplication

$C = A \cdot B$ we compute

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

replace $\min \longrightarrow +$ in Extend-shortest-path
$+ \longrightarrow \cdot$

Then it is a matrix multiplication

$$D^{(m+1)} = D^{m} \cdot W$$

ALL-Shortest-Paths-By-Matrix-Multiplication

$D^{(1)} := W$
for $m = 2$ to $n-1$
  $D^{(m)} :=$ Extend-shortest-path (D^{m-1}, W)
endfor
return $D^{(m-1)}$

$D^{(1)} = W$
$D^{(2)} = W^2$
$D^{(3)} = W^3$
$\vdots$
$D^{(n-1)} = W^{(n-1)}$

$\Rightarrow$

$D^{(1)} = W$
$D^{(2)} = W^2 = W \cdot W$
$D^{(4)} = W^4 = W^2 \cdot W^2$
$D^{(8)} = W^8 = W^4 \cdot W^4$
$\vdots$
$D^{2^{\lceil \log(n-1) \rceil}} = W^{2^{\lceil \log(n-1) \rceil - 1}} \cdot W^{2^{\lceil \log(n-1) \rceil - 1}}$

Since $2^{\lceil \log(n-1) \rceil} \geq n$,
$D^{2^{\lceil \log(n-1) \rceil}} = D^{(n-1)}$

$O(n^3 \log n)$

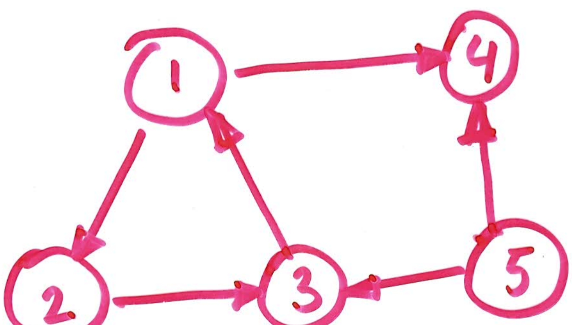
Floyd-Warshall's algorithm.

Transitive closures

Compute

$$d_{ij} = \begin{cases} 1 & \text{if there is a path from } i \text{ to } j \\ 0 & \text{if there is no such path} \end{cases}$$

Ex.



At the kth iteration we get $d_{ij} = 1$ if there is a path from i to j that goes through vertices (excluding i & j) with <u>indices</u> $\leq k$.

	1	2	3	4	5
1	1 ✓	1 ✓	1 °	1 ✓	
2	1 +	1 ✓	1 ✓	1 +	
3	1 ✓	1 *	1 ✓	1 *	
4				1 ✓	
5	1 +	1 +	1 ✓	1 ✓	1 ✓

✓ Initialization

* k=1 (3,4), (3,2)

° k=2 (1,3)

\+ k=3 (2,4), (2,1)
(5,1), (5,2)

k=4 no change

k=5 no change

```
for k:=1 to n do
  for i:=1 to n do
    for j:=1 to n do
      if d_ik = 1 and d_kj = 1 then
        d_ij = 1
      endif
    endfor
  endfor
endfor
```

$O(n^3)$

Floyd-Warshall

rule: At the k^{th} iteration, compute the length of the shortest path from i to j that contains only nodes $\leq k$ (excluding i & j).

```
D = W;
for k:=1 to n do
  for i:=1 to n do
    for j:=1 to n do
      if d_ik + d_kj < d_ij then
        d_ij := d_ik + d_kj
      endif
    endfor
  endfor
endfor
```

$O(n^3)$