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 edge weights, no negative cycle)

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

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

Input: Adjacency matrix $W$.

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.

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

$$d_{ij}^{(m)} = \min \left\{ d_{ij}^{(m-1)}, \min_{1 \leq k \leq n} \left\{ d_{ik}^{(m-1)} + w_{kj} \right\} \right\} \quad \text{for } m \geq 2$$

$$d_{ij}^{(n)} = d_{ij}^{(n-1)} = d_{ij}^{(n)} = \ldots$$

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

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

for $i := 1$ to $n$
  for $j := 1$ to $n$
    $d_{ij}^{(m+1)} := \infty$
  endfor
endfor

for $k := 1$ to $n$
  $d_{ij}^{(m+1)} := \min \{ d_{ij}^{(m+1)}, d_{ik}^{(m)} + w_{kj} \}$
endfor
endfor

return $D^{m+1}$
For matrix multiplication

\[ C = A \cdot B \text{ we compute} \]

\[ C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}. \]

Replace \( \min \rightarrow + \) in Extend-shortest-path

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 \( n-1 \)

\[ D^{(m)} := \text{Extend-shortest-path}(D^{m-1}, W) \]

endfor

Return \( D^{(m-1)} \)

\[ D^{(1)} = W \]

\[ D^{(2)} = W^2 \]

\[ D^{(3)} = W^3 \]

\[ \Rightarrow \]

\[ D^{(n)} = \text{W}^{\lfloor \log(n-1) \rfloor} \cdot \text{W}^{(n-1) \cdot \log(n-1)} \cdot W^{2 \log(n-1)} \cdot W \]

Since \( 2^{\lfloor \log(n-1) \rfloor} \geq n \),

\[ D \approx D^{(n-1)} \]

\( O(n^3 \log n) \)
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 k-th iteration we get \( d_{ij} = 1 \) if there is a path from \( i \) to \( j \) that goes through vertices (excluding \( i \) and \( j \)) with indices \( \leq k \).

- 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

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

\( O(n^3) \)