

Floyd-Warshall Algorithm

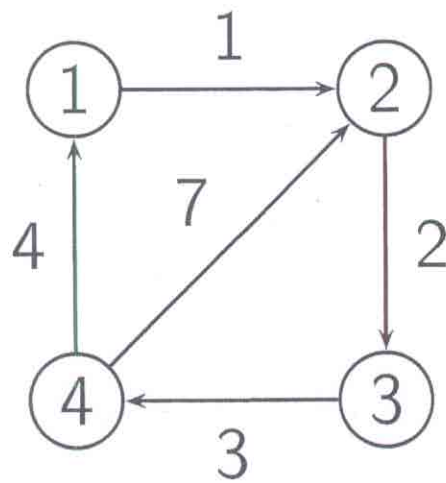
- Finds shortest paths between all pairs of nodes
- $d_{i,j} \geq 0$, but no negative cycles
- $d_{i,k} \geq d_{i,j} + d_{j,k}$

$$d_{i,k} \leftarrow \min(d_{i,k}, d_{i,j} + d_{j,k})$$

For $j = 1, 2, \dots, n$ and all $i, k \neq j$

After $j = n$, we have the shortest distance between any pair of nodes.

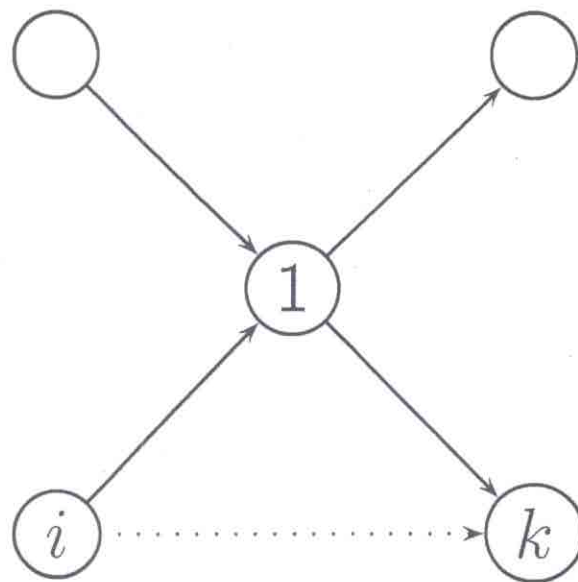
Numerical Example



	①	②	③	④
①	0	1	∞	∞
②	∞	0	2	∞
③	∞	∞	0	3
④	4	7	∞	0

Another Example

When $j = 1 \dots$

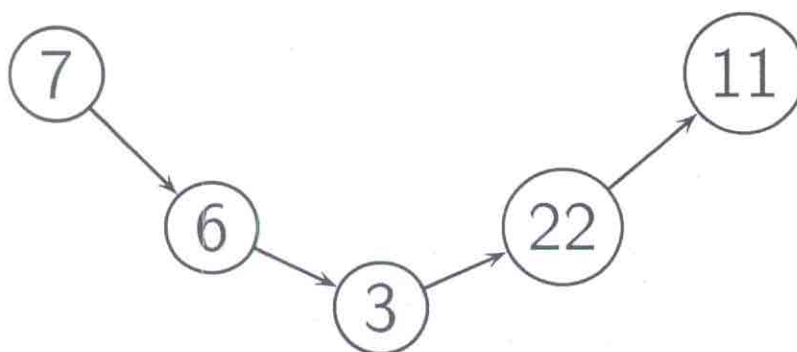


If $d_{i,k} \leq d_{i,1} + d_{1,k}$ then $d_{i,k}$ remains unchanged

If $d_{i,k} > d_{i,1} + d_{1,k}$ then $d_{i,k}$ is updated by the
sum

Then try $j = 2$, update with smaller values

Basic Arcs

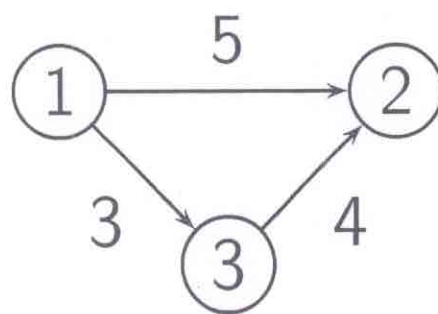


If $(7) \sim (11)$ is shortest
then $(6) \sim (22)$ is shortest
then $(6) \sim (3)$ is shortest

Basic Arc (i, j) : An arc (i, j) is a basic arc iff
the shortest path from (i) to (j) is the arc
 (i, j) .

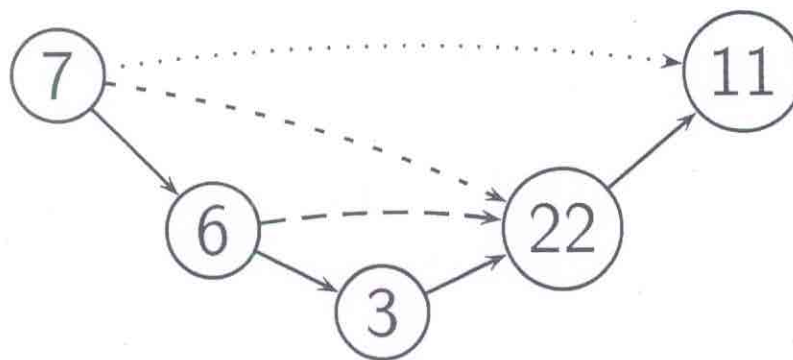
-
- ☐ A shortest path must consist of basic arcs?
 - ☐ A path consisting of basic arcs must be shortest?

Counterexample showing that a path consisting of basic arcs is not necessarily shortest.

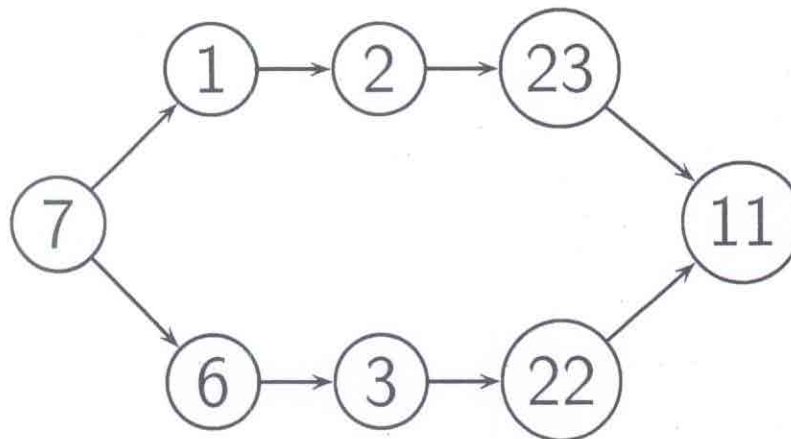


Proof that a shortest path must consist of basic arcs.

If we focus on an arbitrary shortest path, and we can get the shortest distance using basic arcs, then it is correct for all



Ties



In case of tie, which path will be picked by the computer?

We want the path also

We got the shortest distance, but not the shortest path

$p_{i,k} = k$ for all i, k to start

We have an associated array:

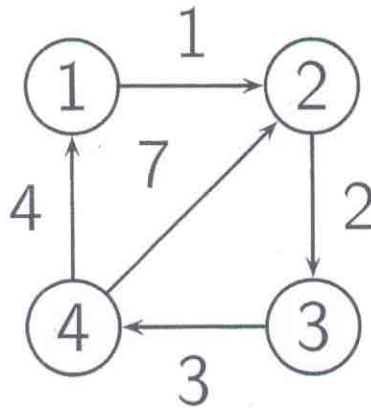
	①	②	③	④
①	1	2	3	4
②	1	2	3	4
③	1	2	3	4
④	1	2	3	4

$$p_{i,k} = \begin{cases} j & \text{if } d_{i,k} > d_{i,j} + d_{j,k} \\ \text{same} & \text{if } d_{i,k} \leq d_{i,j} + d_{j,k} \end{cases}$$

This gives us *some* node j between i and k ,
but we want the *first* node between i and k .

Change the algorithm to:

$$p_{i,k} = \begin{cases} p_{i,j} & \text{if } d_{i,k} > d_{i,j} + d_{j,k} \\ \text{same} & \text{if } d_{i,k} \leq d_{i,j} + d_{j,k} \end{cases}$$



$$j = 1 : p_{4,2} \leftarrow p_{4,1} \leftarrow 1$$

$$j = 2 : p_{1,3} \leftarrow p_{4,2} \leftarrow p_{4,1} \leftarrow 1$$

$$j = 3 : p_{4,3} \leftarrow p_{4,2} \leftarrow 1$$

$$\textcircled{4} \sim \textcircled{3} = p_{4,3} = \textcircled{1}$$

$$\textcircled{1} \sim \textcircled{3} = p_{1,3} = \textcircled{2}$$

$$\textcircled{2} \sim \textcircled{3} = p_{2,3} = \textcircled{3}$$

$d_{i,k}:$

	①	②	③	④
①	0	1	∞	∞
②	∞	0	2	∞
③	∞	∞	0	3
④	4	7	∞	0

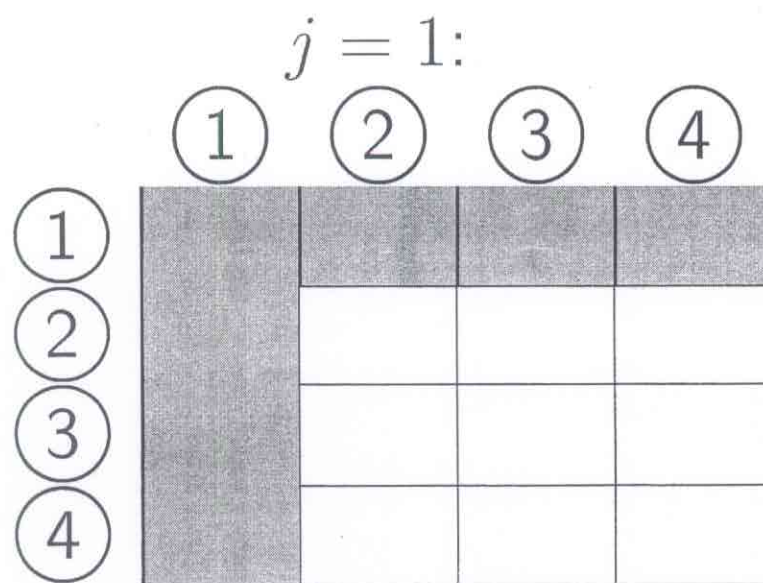
$j = 1:$

	①	②	③	④
①				
②				
③				
④		5		

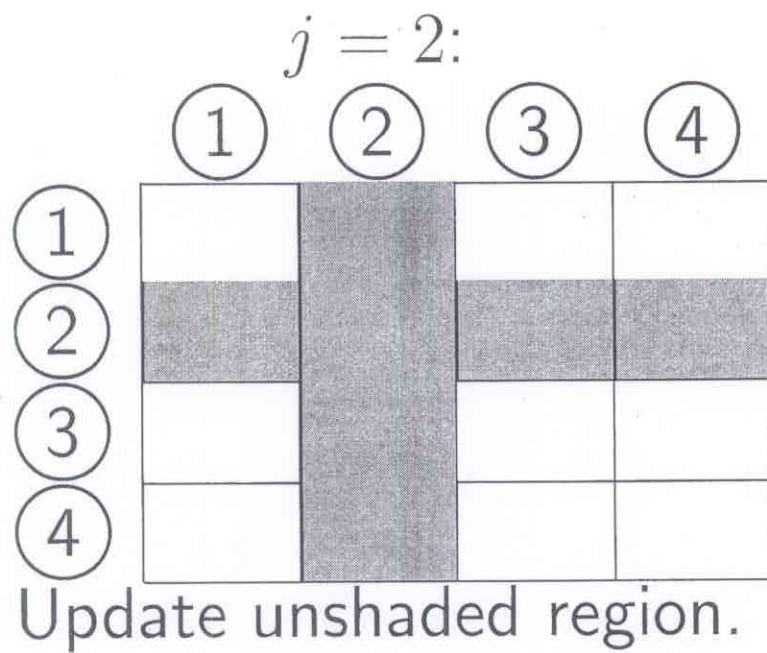
$$p_{4,2} \leftarrow p_{4,1} = 1$$

Implementation

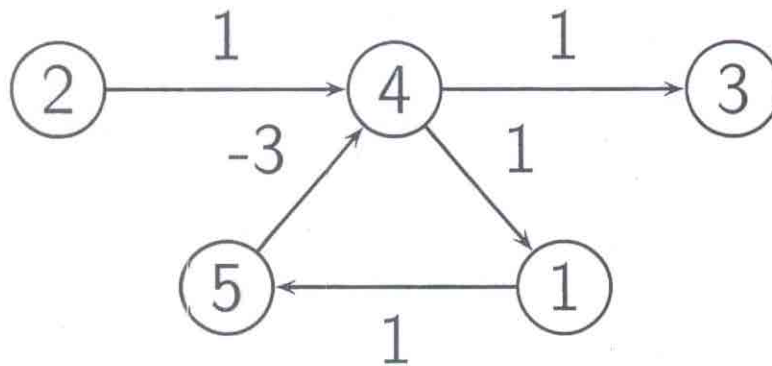
Given n by n array



Update $(n - 1)$ by $(n - 1)$ unshaded region.



Negative cycles?



How to find negative cycle?

Set $d_{i,i} = \infty$ to start, then run Floyd Warshall

