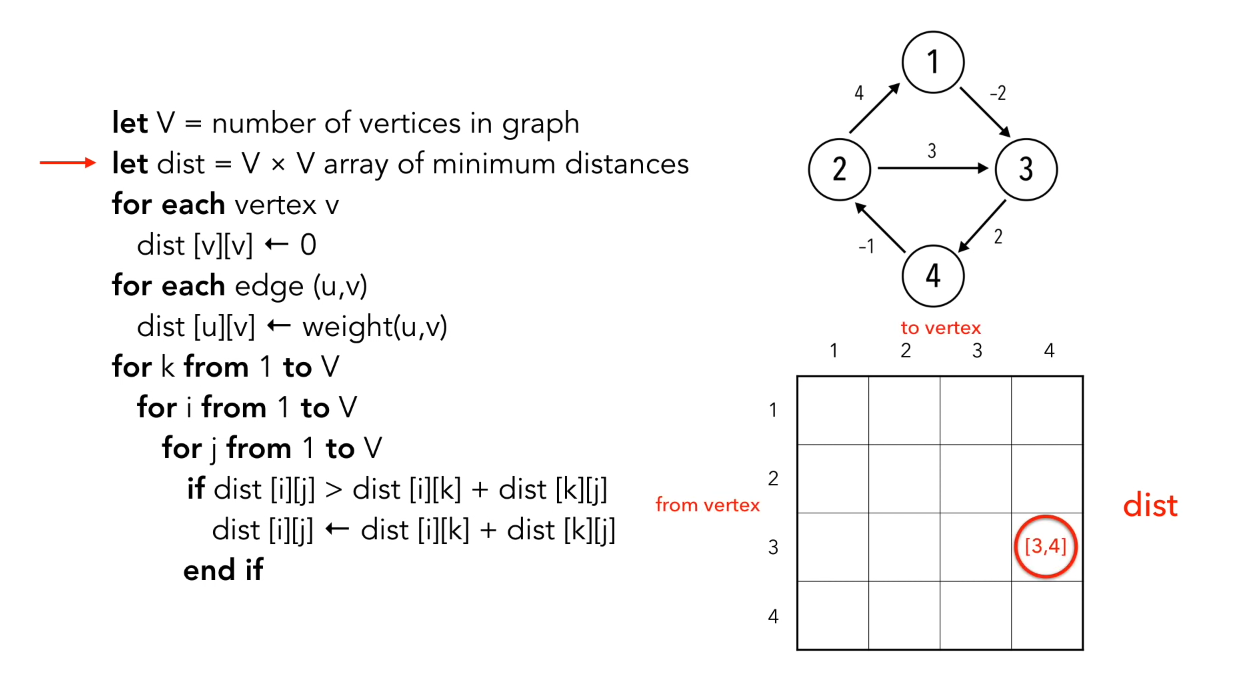
# Floyd-Warshall

The algorithm finds the shortest path between all vertices. Negative cycles are not allowed.

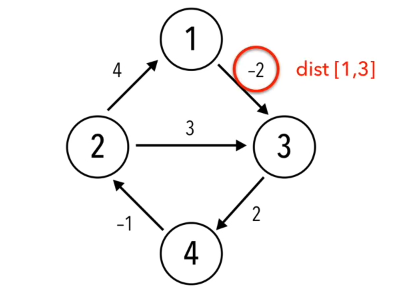


The distance array has the size of all vertices \* all vertices.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | 0 |  |  |  |
| 2 |  | 0 |  |  |
| 3 |  |  | 0 |  |
| 4 |  |  |  | 0 |

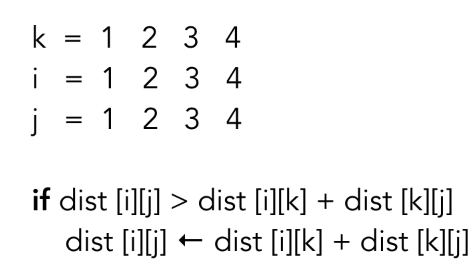
First the distance from the vertex itself to itself is found. In this case it is 0. If there was loops it would be different.

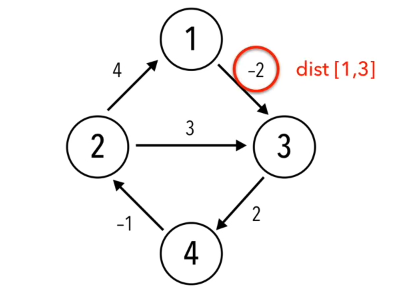
Afterwards all the simple edges are analyzed and their distance is put into the distance array.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | 0 |  | -2 |  |
| 2 | 4 | 0 | 3 |  |
| 3 |  |  | 0 | 2 |
| 4 |  | -1 |  | 0 |

If we know look at the pseudo code, we can see that





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | 0 |  | -2 |  |
| 2 | 4 | 0 | 3 |  |
| 3 |  |  | 0 | 2 |
| 4 |  | -1 |  | 0 |

If we look at the if statement, and take the first iteration of the loops it is:

dist[1][1] > dist[1][1] + dist[1][1] = 0 > 0 + 0   
this is **false** and the array is not updated.

With j = 2(it is the last loop) we get

dist[1][2] > dist[1][1] + dist[1][2] = ∞ > 0 + ∞   
which is **false** again

When I = and j = 3 we get

dist[2][3] > dist[2][1] + dist[1][3] = 3 > 4 + -2  
which is **true** and the array is updated to:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | 0 |  | -2 |  |
| 2 | 4 | 0 | 2 |  |
| 3 |  |  | 0 | 2 |
| 4 |  | -1 |  | 0 |

Link til youtube: <https://www.youtube.com/watch?v=4OQeCuLYj-4>