

Objectives

Floyd Warshall

Dr. Mattox Beckman

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
DEPARTMENT OF COMPUTER SCIENCE

Your Objectives:

- ▶ Implement the Floyd Warshall algorithm.
- ▶ Determine when this algorithm is “safe” to run.

The Algorithm

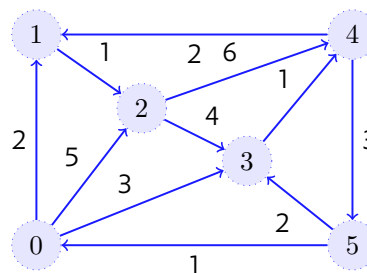
```

0 // From Competitive Programming 3
1 for (int k = 0; k < V; k++)
2   for (int i = 0; i < V; i++)
3     for (int j = 0; j < V; j++)
4       AdjMat[i][j] = min(AdjMat[i][j], AdjMat[i][k] + AdjMat[k][j]);

```

- ▶ Each run of the k loop adds node k to our shortest path collection.
- ▶ On today's machines, this can work up to $V = 500$.

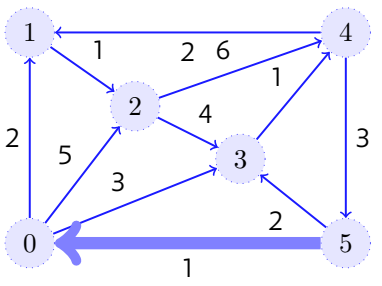
An Example



$k = -1$:

	0	1	2	3	4	5
0	0	2	5	9	∞	∞
1	∞	0	1	∞	∞	∞
2	∞	∞	0	4	6	∞
3	∞	∞	∞	0	1	∞
4	2	∞	∞	∞	0	3
5	1	∞	∞	2	∞	0

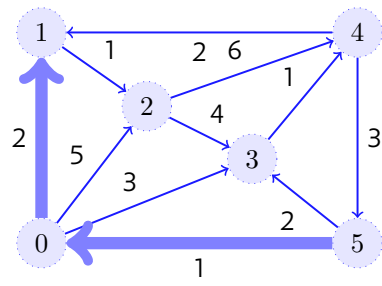
An Example



k = -1:

	0	1	2	3	4	5
0	0	2	5	9	∞	∞
1	∞	0	1	∞	∞	∞
2	∞	∞	0	4	6	∞
3	∞	∞	∞	0	1	∞
4	2	∞	∞	∞	0	3
5	1	∞	∞	2	∞	0

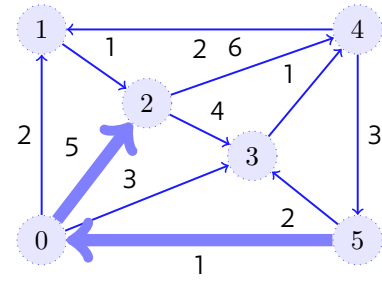
An Example



k = -1:

	0	1	2	3	4	5
0	0	2	5	9	∞	∞
1	∞	0	1	∞	∞	∞
2	∞	∞	0	4	6	∞
3	∞	∞	∞	0	1	∞
4	2	∞	∞	∞	0	3
5	1	3	∞	2	∞	0

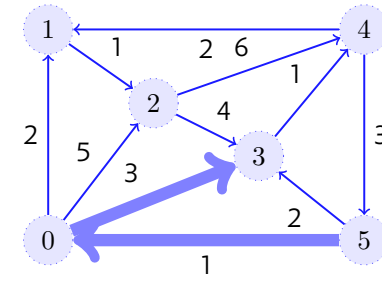
An Example



k = -1:

	0	1	2	3	4	5
0	0	2	5	9	∞	∞
1	∞	0	1	∞	∞	∞
2	∞	∞	0	4	6	∞
3	∞	∞	∞	0	1	∞
4	2	∞	∞	∞	0	3
5	1	3	6	2	∞	0

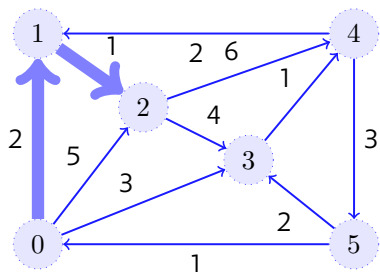
An Example



k = -1:

	0	1	2	3	4	5
0	0	2	5	9	∞	∞
1	∞	0	1	∞	∞	∞
2	∞	∞	0	4	6	∞
3	∞	∞	∞	0	1	∞
4	2	∞	∞	∞	0	3
5	1	3	6	2	∞	0

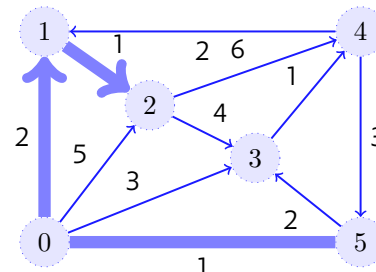
Example, k = 1



k = 0

	0	1	2	3	4	5
0	0	2	3	9	∞	∞
1	∞	0	1	∞	∞	∞
2	∞	∞	0	4	6	∞
3	∞	∞	∞	0	1	∞
4	2	4	5	11	0	3
5	1	3	6	2	∞	0

Example, k = 1



k = 0

	0	1	2	3	4	5
0	0	2	3	9	∞	∞
1	∞	0	1	∞	∞	∞
2	∞	∞	0	4	6	∞
3	∞	∞	∞	0	1	∞
4	2	4	5	11	0	3
5	1	3	4	2	∞	0

Printing the Paths

```

0 for (int i = 0; i < V; i++)
1   for (int j = 0; j < V; j++)
2     p[i][j] = i;
3 for (int k = 0; k < V; k++)
4   for (int i = 0; i < V; i++)
5     for (int j = 0; j < V; j++)
6       if (AdjMat[i][k] + AdjMat[k][j] < AdjMat[i][j]) {
7         AdjMat[i][j] = AdjMat[i][k] + AdjMat[k][j];
8         p[i][j] = p[k][j];
9       }
10 void printPath(int i, int j) {
11   if (i != j) printPath(i, p[i][j]);
12   printf("%d", j);
13 }

```