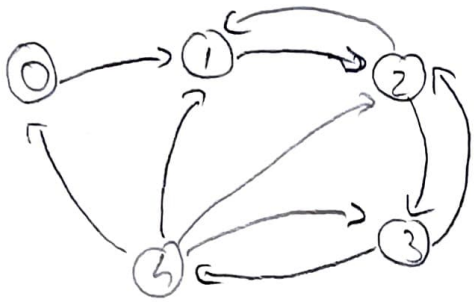
Lab 2 - Problem 1



0 : 1
1 : 2
2 : 1, 3
3 : 2, 4
4 : 0, 1, 2, 3

	x	y	g	visited	prev
initialization			0	0 1 2 3 4 T F F F F	0 1 2 3 4
iteration 1 iteration 1.1	0	1	= 1	0 1 2 3 4 T T F F F	0 1 2 3 4 0
iteration 2 iteration 2.1	1	2	= 2	0 1 2 3 4 T T T F F	0 1 2 3 4 0 1
iteration 3 iteration 3.1 iteration 3.2	2	1 3	= = 3	0 1 2 3 4 T T T F F T T T T F	0 1 2 3 4 0 1 0 1 2
iteration 4 iteration 4.1 iteration 4.2	3	2 4	= = 4	0 1 2 3 4 T T T T F T T T T T	0 1 2 3 4 0 1 2 0 1 2 3
iteration 5 iteration 5.1 iteration 5.2 iteration 5.3 iteration 5.4	4	0 1 2 3	= = = = =		

The queue is empty, so our program stops and it returns the prev

0 1 2 3 4
0 1 2 3

In the function print_path we get as a parameter the prev list and we build the path.

end_point = 4 prev[4] = 3 prev[3] = 2 prev[2] = 1 prev[1] = 0

=> path {0, 1, 2, 3, 4} after reversing len(path) - 1 = **4**

graph 1k :

1 - 100 : The length of the shortest path is 6.

{ 1, 5, 487, 175, 699, 624, 100 }

100 - 1 : The length of the shortest path is 5.

{ 100, 416, 354, 865, 109, 1 }

graph 10k

1 - 100 : The length of the shortest path is 8.

{ 1, 3300, 2607, 523, 6211, 5359, 9794, 5173, 100 }

100 - 1 : The length of the shortest path is 7.

{ 100, 2398, 3054, 5232, 8217, 2478, 7151, 1 }

graph 100k

1 - 100 : The length of the shortest path is 8.

{ 1, 17024, 27471, 14969, 3075, 4156, 32753, 14973, 100 }

100 - 1 : The length of the shortest path is 8.

{ 100, 44340, 54527, 6606, 53263, 95930, 98655, 58288, 1 }