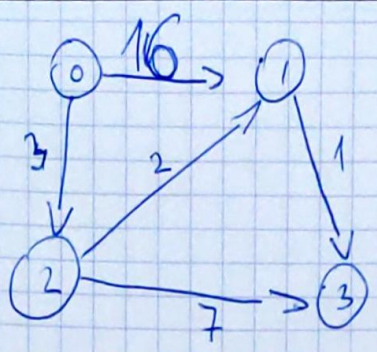
Graph:



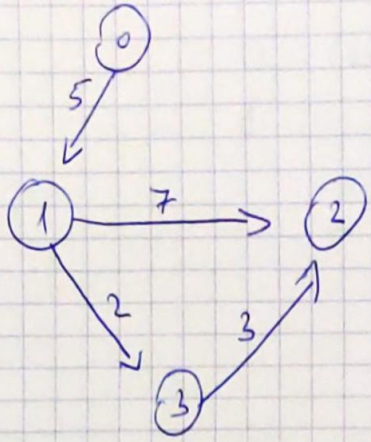
source: 0
dest: 3

	changed	edges	current Map	prev Map
init	true		0 1 2 3 0 - - -	
iter 1	false			
	true	0 → 1	0 16 - -	0 1 2 3 0 - - -
	true	0 → 2	0 16 3 -	
	true	1 → 3	0 16 3 17	
	true	2 → 1	0 5 3 6	
	~~false~~	2 → 3	0 5 3 6	
iter 2	false		0 1 2 3	0 1 2 3
		0 → 1	0 5 6 3	0 16 - -
		0 → 2	0 5 6 3	0 16 3 -
		1 → 3	0 5 6 3	0 16 3 17
		2 → 1	0 5 6 3	0 5 3 6
		2 → 3	0 5 6 3	0 5 3 6

To get the path, we go backwards from target and use it's inbound neighbours to ~~see how we got~~ determine the last node that got us the. We repeat this process `length` times.

Path: $0 \xrightarrow{3} 2 \xrightarrow{2} 1 \xrightarrow{1} 3$ cost: 6

Graph



source: 0
dest: 2

	changed	edges	currentMap: dict	prevMap: dict
init	true		0 1 2 3 0 - - -	
iter 1	false			
	true	0→1	0 5 - -	0 1 2 3 0 - - -
	true	1→2	0 5 ~~12~~ -	
	true	1→3	0 5 ~~12~~ 7	
	true	3→2	0 5 10 7	
iter 2	false			0 1 2 3
		0→1	0 5 ~~10~~ ~~7~~	0 5 - -
		1→2	0 5 10 7	0 5 12 -
		1→3	0 5 10 7	0 5 12 7
		3→2	0 5 10 7	0 5 10 7

To get the path, we go back from target and check it's inbound neighbours to see which one gives the value currently in currentMap.

Path: $0 \xrightarrow{5} 1 \xrightarrow{2} 3 \xrightarrow{3} 2$ cost: 10