Directed graph with 8 nodes and 12 edges

The input file is:

8 12
5 7 11
3 4 21
4 7 56
6 0 80
1 7 76
5 2 10
5 1 6
0 4 68
6 1 18
6 4 94
2 3 95
2 7 52

Dictionary_cost = {(5, 7): 11, (3, 4): 21, (4, 7): 56, (6, 0): 80, (1, 7): 76, (5, 2): 10, (5, 1): 6, (0, 4): 68, (6, 1): 18, (6, 4): 94, (2, 3): 95, (2, 7): 52}

Dictionary_in = {0: [6], 1: [5, 6], 2: [5], 3: [2], 4: [3, 0, 6], 5: [], 6: [], 7: [5, 4, 1, 2]}

Dictionary_out = {0: [4], 1: [7], 2: [3, 7], 3: [4], 4: [7], 5: [7, 2, 1], 6: [0, 1, 4], 7: []}

Topological sorting using the DFS algorithm

→ consider self to be the graph

calls	vertex, other_vertex	inProcess	fullyProcessed	sortedList	ob
Initialisation		14	14	[]	
self.topo_sort_dfs(0, [], 14, 14)	vertex = 0	14, 14, 104	14	[]	[]
	other_vertex = 6				
self.topo_sort_dfs(6,[],14,14,104)	vertex = 6	204, 10,64, 204	14, 264	[]	[6]	true
	vertex = 0	14, 14	16,04	[6,0]	true
self.topo_sort_dfs(1,[6,0],16,04,14)	vertex = 1	14, 214	16,04	[6,0]	true
	other_vertex = 5				
self.topo_sort_dfs(5,[6,0],16,04,14)	vertex = 5	214, 34,54, 214	16,04	[6,0]	
	vertex = 1	14	16,0,54	[6,0,5]	true
self.topo_sort_dfs(2,[6,0,5,1],16,0,5,14,16,0,5,14,44)	vertex = 1	14	16,0,5,14	[6,0,5,1]	true
	vertex = 2	14, 24, 14	16,0,5,14, 16,0,5,1,24	[6,0,5,1]	true
self.topo_sort_dfs(3,16,0,5,1,2], 16,0,5,1,24,14)	vertex = 3	14, 34, 14	16,0,5,1,24, 16,0,5,1,2,34	[6,0,5,1,2]	true
self.topo_sort_dfs(4,16,0,5,1,2,3], 16,0,5,1,2,34, 14)	vertex = 4	14, 44, 14	16,0,5,1,2,34, 16,0,5,1,2,3,44	[6,0,5,1,2,3]	true
self.topo_sort_dfs(7,16,0,5,1,2,34,16,0,5,1,2,3,44,14)	vertex = 7	14, 74, 14	16,0,5,1,2,3,44, 16,0,5,1,2,3,4,74	[6,0,5,1,2,3,4]	true
			16,0,5,1,2,3,4,74	[6,0,5,1,2,3,4,7]	true

→ if the graph is not a DAG, the returned value will be false and the algorithm will consider the sortedList empty

topological order

Highest_cost_path: self_graph.self_current].highest_cost_path(5,4): topological_order_list = [6,0,5,1,2,3,7]

	vertex	other_vertex	dist_dictionary									prev_dictionary							
			0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
initialization			$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	-1	-1	-1	-1	-1	-1	-1	-1
iteration 1	6	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	-1	-1	-1	-1	-1	-1	-1	-1
		1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	-1	-1	-1	-1	-1	-1	-1	-1
		4	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	-1	-1	-1	-1	-1	-1	-1	-1
iteration 2	0	4	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	-1	-1	-1	-1	-1	-1	-1	-1
iteration 3	5	7	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	11	-1	-1	-1	-1	-1	5	-1	5
		2	$-\infty$	$-\infty$	10	$-\infty$	$-\infty$	0	$-\infty$	11	-1	-1	5	-1	-1	5	-1	5
		1	$-\infty$	6	10	$-\infty$	$-\infty$	0	$-\infty$	11	-1	5	5	-1	-1	5	-1	5
iteration 4	1	4	$-\infty$	6	10	$-\infty$	0	0	$-\infty$	82	-1	5	5	-1	-1	5	-1	1
iteration 5	2	3	$-\infty$	6	10	105	0	0	$-\infty$	82	-1	5	5	2	-1	5	-1	1
		7	$-\infty$	6	10	105	0	0	$-\infty$	82	-1	5	5	2	-1	5	-1	1
iteration 6	3	4	$-\infty$	6	10	105	126	0	$-\infty$	82	-1	5	5	2	3	-1	-1	1

After iteration 6 we stop because vertex = 4 = vertex_end => dist[4]=126

The highest cost path from vertex 5 to vertex 4 has the cost: dist[4]=126 and it is built backwards from the prev dictionary:

vertex_end=4, prev[4]=3, prev[3]=2, prev[2]=5 => path: $5 \xrightarrow{10} 2 \xrightarrow{95} 3 \xrightarrow{21} 4$