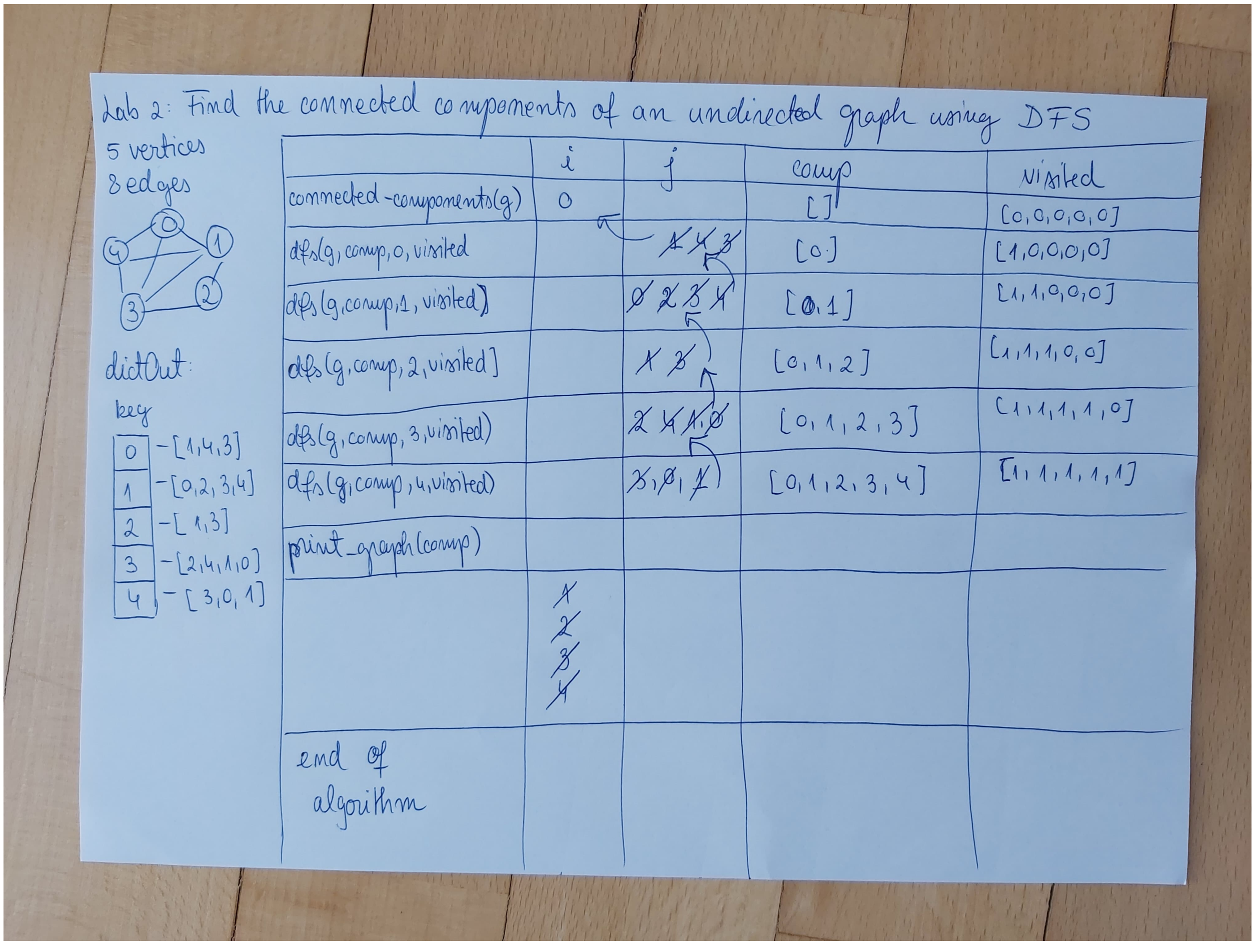


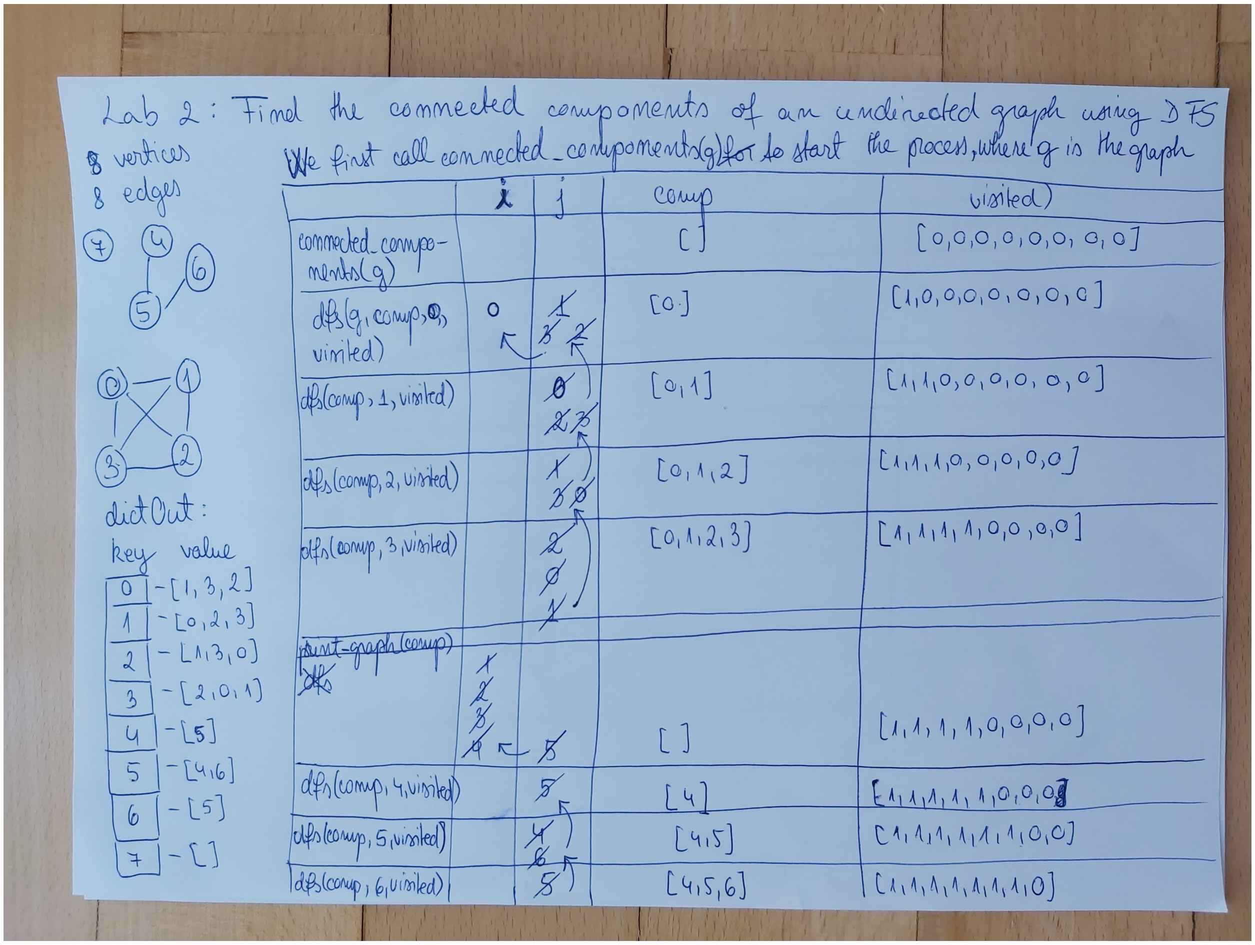
Lab 2: Find the connected components of an undirected graph using DFS

5 vertices
8 edges

dictOut:

key	
0	- [1, 4, 3]
1	- [0, 2, 3, 4]
2	- [1, 3]
3	- [2, 4, 1, 0]
4	- [3, 0, 1]

	i	j	comp	visited
connected-components(g)	0		[ ]	[0, 0, 0, 0, 0]
dfs(g, comp, 0, visited		~~1~~ ~~4~~ ~~3~~	[0.]	[1, 0, 0, 0, 0]
dfs(g, comp, 1, visited)		~~0~~ ~~2~~ ~~3~~ ~~4~~	[0, 1]	[1, 1, 0, 0, 0]
dfs(g, comp, 2, visited]		~~1~~ ~~3~~	[0, 1, 2]	[1, 1, 1, 0, 0]
dfs(g, comp, 3, visited)		~~2~~ ~~4~~ ~~1~~, ~~0~~	[0, 1, 2, 3]	[1, 1, 1, 1, 0]
dfs(g, comp, 4, visited)		~~3~~, ~~0~~, ~~1~~	[0, 1, 2, 3, 4]	[1, 1, 1, 1, 1]
print_graph(comp)				
	~~1~~ ~~2~~ ~~3~~ ~~4~~			
end of algorithm				



Lab 2: Find the connected components of an undirected graph using DFS

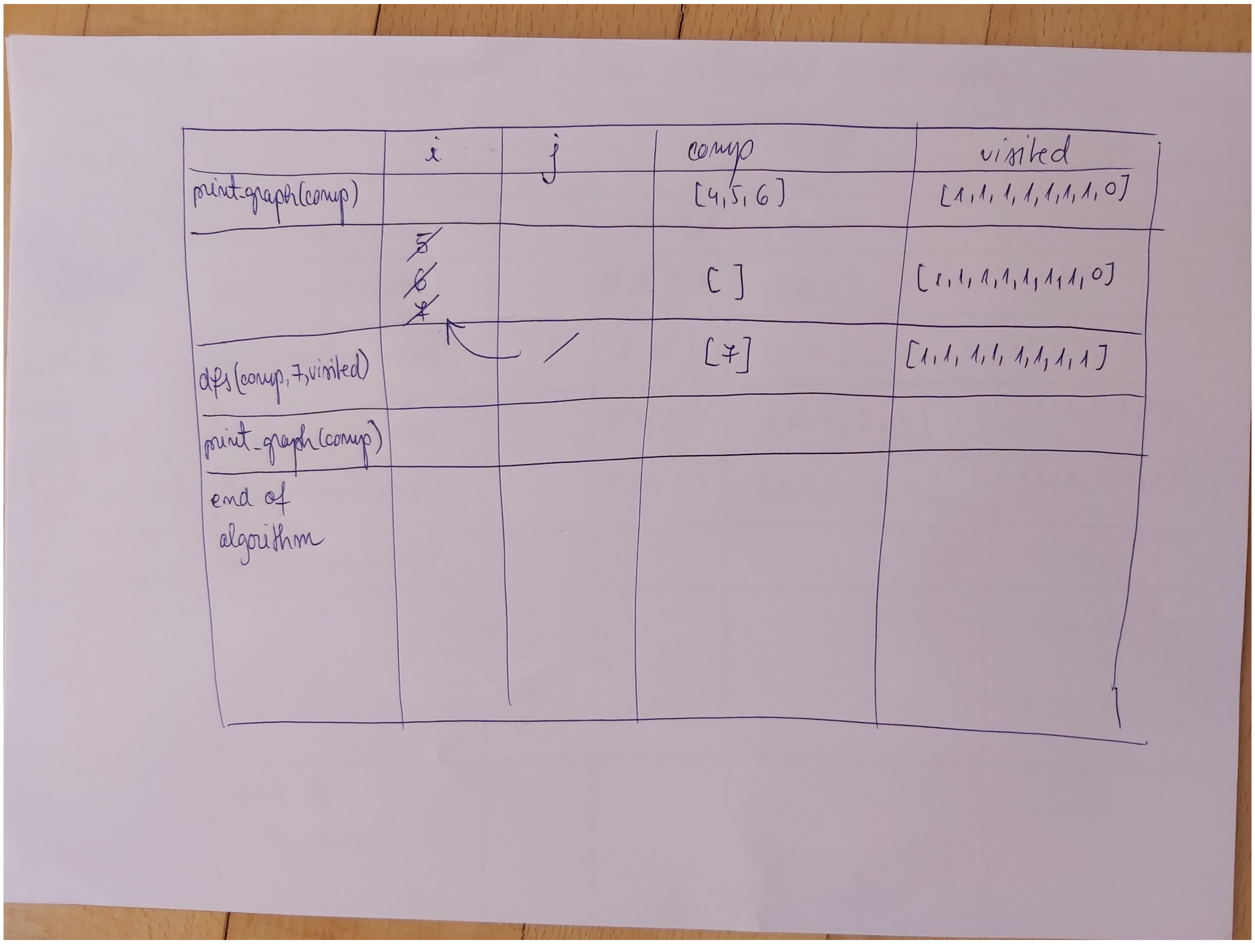
8 vertices
8 edges

We first call connected_components(g) ~~for~~ to start the process, where g is the graph

dictOut:

key	value
0	- [1, 3, 2]
1	- [0, 2, 3]
2	- [1, 3, 0]
3	- [2, 0, 1]
4	- [5]
5	- [4, 6]
6	- [5]
7	- [ ]

	i	j	comp	visited)
connected_components(g)			[ ]	[0, 0, 0, 0, 0, 0, 0, 0]
dfs(g, comp, 0, visited)	0	~~1~~ ~~3~~ ~~2~~	[0.]	[1, 0, 0, 0, 0, 0, 0, 0]
dfs(comp, 1, visited)		~~0~~ ~~2~~ ~~3~~	[0, 1]	[1, 1, 0, 0, 0, 0, 0, 0]
dfs(comp, 2, visited)		~~1~~ ~~3~~ ~~0~~	[0, 1, 2]	[1, 1, 1, 0, 0, 0, 0, 0]
dfs(comp, 3, visited)		~~2~~ ~~0~~ ~~1~~	[0, 1, 2, 3]	[1, 1, 1, 1, 0, 0, 0, 0]
~~print-graph(comp)~~ ~~dfs~~	~~1~~ ~~2~~ ~~3~~ ~~4~~	~~5~~	[ ]	[1, 1, 1, 1, 0, 0, 0, 0]
dfs(comp, 4, visited)		~~5~~	[4]	[1, 1, 1, 1, 1, 0, 0, 0]
dfs(comp, 5, visited]		~~4~~ ~~6~~	[4, 5]	[1, 1, 1, 1, 1, 1, 0, 0]
dfs(comp, 6, visited)		~~5~~	[4, 5, 6]	[1, 1, 1, 1, 1, 1, 1, 0]



	i	j	comp	visited
print_graph(comp)			[4,5,6]	[1,1,1,1,1,1,1,0]
	~~5~~ ~~6~~ ~~7~~		[ ]	[1,1,1,1,1,1,1,0]
dfs(comp,7,visited)		/	[7]	[1,1,1,1,1,1,1,1]
print_graph(comp)				
end of algorithm				