



Initially we assume Everything is safe Node:

0	1	2	3	4	5	6
T	T	T	T	T	T	T

Visited:

0	1	2	3	4	5	6

Ancestor:

0	1	2	3	4	5	6

If ancestor = True, then
we are in a cycle

Dfs(0):

0	1	2	3	4	5	6
T						

visited

0	1	2	3	4	5	6
T						

ancestor

dps(1)

0	1	2	3	4	5	6
T	T					

visited

ancestor

dps(2)

0	1	2	3	4	5	6
T	T	T				

visited

ancestor

↓

$dp(5)$	0 1 2 3 4 5 6	Visited
	T T T T	
	0 1 2 3 4 5 6	Ancestor
	T T T T	

since 5 has No Neighbor

- * We mark it as safe Node
- * And uncheck ancestor, as it has finished its exploration.
- * return to Node ②

Node 2 finished Exploration

- * all its neighbors are safe Node
so Node 2 is also safe Node

* return to Node ①

↓

$dp(3)$	0 1 2 3 4 5 6	Visited
	T T T T T	
	T T F T F	Ancestor

→ Neighbor of Node 3 is Node 0

- * Node 0 → is ancestor of Node 3
- * Hence there is a cycle & is Not a safe Node
- * Uncheck ancestor
- * return to Node ①

Node 1 finished Exploration

- Node 2 is safe

- Node 3 is unsafe

* so Node 1 is unsafe

* return to Node 0.

|| Node 0

djs(2)

0	1	2	3	4	5	6
T	T	T	T	T	T	

Visited

* Already Visited

T	F	F	F	F	
---	---	---	---	---	--

ancestor

* Is not ancestor

* Is safe Node →

	F	T	F		T
--	---	---	---	--	---

Node(1) is unsafe Node

So, Node(0) is unsafe Node

Djs(4)

0	1	2	3	4	5	6
T	T	T	T	T	T	

Visited

0	1	2	3	4	5	6
F	F	F	F	T	F	

ancestor

Node(5)

* already Visited

* Not ancestor

* is safe Node

so Node 4 is safe Node

Node 6 has No Neighbor \rightarrow hence Sane Node.

so finally the Sane Nodes are

0	1	2	3	4	5	6
F	F	T	F	T	T	T

[2, 4, 5, 6]