

Important lecture

Strongly Connected Component.

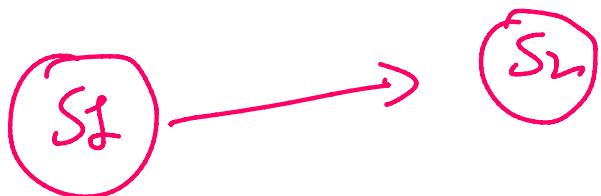
Kosaraju Algorithm.

- ① Topological Sort
- ② Reverse the Graph
- ③ Do the DFS Traversal.

For finding SCC, we have to traverse the graph from the node that is going to be finished first



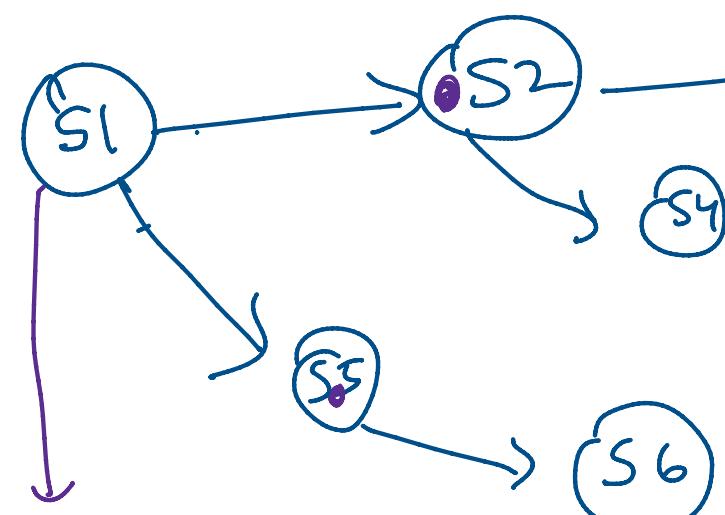
e.g.:
node that is ...



But we can't say surely that all the nodes in S_2 will have lesser finishing time as of all the nodes in S_1 , so if there is any node in S_1 who has lesser finishing time as of nodes in S_2 , it will not give me correct answer.

Solutions

The one thing I can surely say that there must exist a node in S_1 who has higher finishing time as of the nodes in S_2 .



There exist a node in this component with higher finishing time.

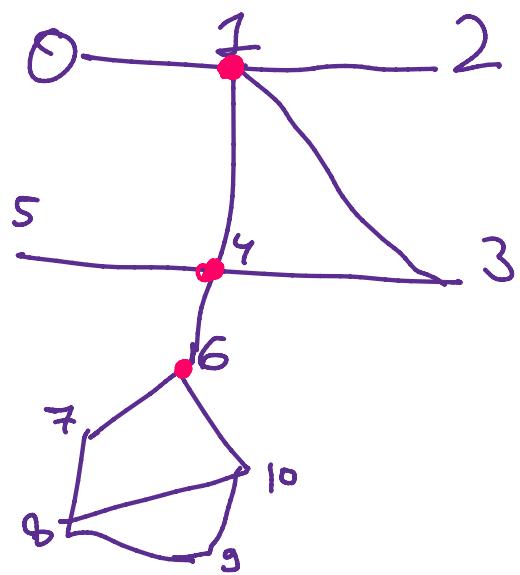
reverse of graph $\underline{\underline{G}}$
the node which has higher finishing time will be treated as the node with $-\infty$.

finishing time

the node with
least finishing
time in G'.
i.e.

Articulation Point:

A Point by removing that
Number of Components increase.



→ Articulation
Point.

Input Graph
G



