

Algorithme de Tarjan

Procédure principale

```
CFC()=  
  date ← 0  
  DEBUT ← [0, ..., 0]  
  P ← Pile_Vide()  
  Ncfc ← 0  
  CFC ← [0, ..., 0]  
  pour tout i ∈ S  
    si DEBUT[i]=0 alors TARJAN(i)  
  renvoie CFC
```

Algorithme de Tarjan

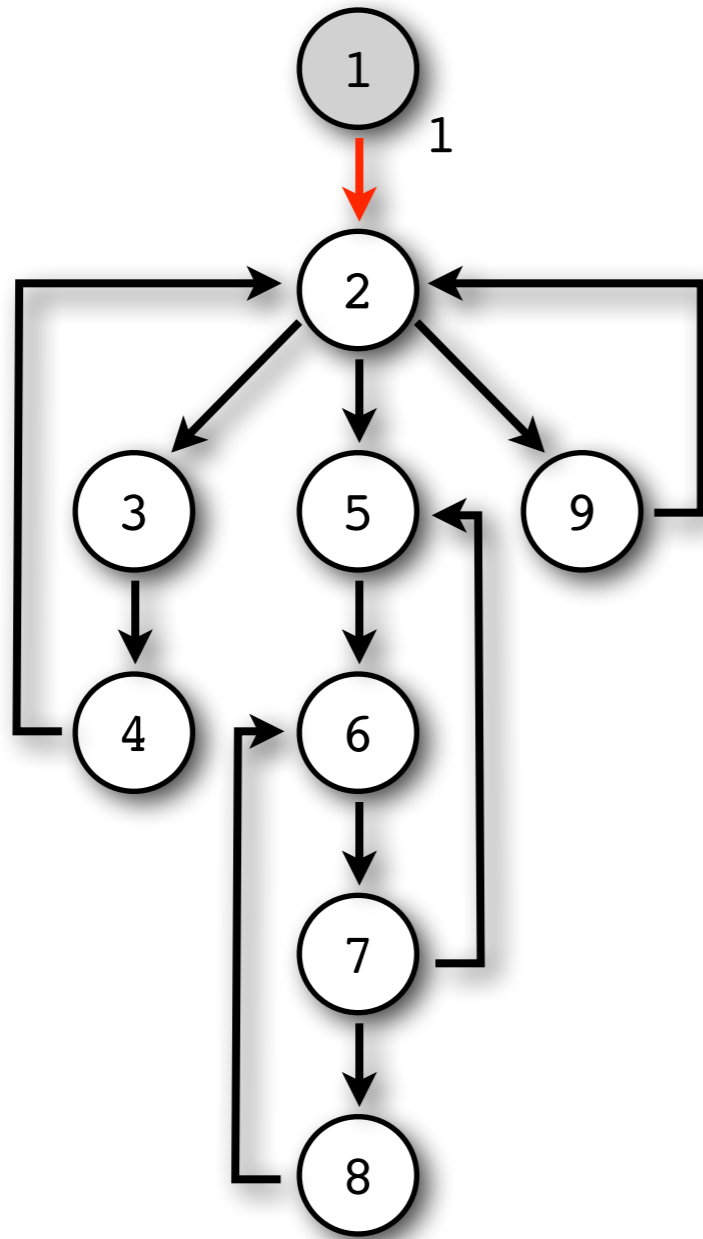
Procédure récursive

```
TARJAN(i)=
  date ← date + 1
  DEBUT[i] ← date
  min ← DEBUT[i]
  Empile(P,i)
  pour tout j∈Adj[i]
    si DEBUT[j]=0 alors min ← MIN(min,TARJAN(j))
    sinon si CFC[j]=0 alors min ← MIN(min,DEBUT[j])
  si min=DEBUT[i] alors
    Ncfc ← Ncfc + 1
    répète
      k ← Depile(P)
      CFC[k] ← Ncfc
    tant que k≠i
  renvoie min
```

Algorithme de Tarjan

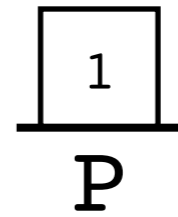
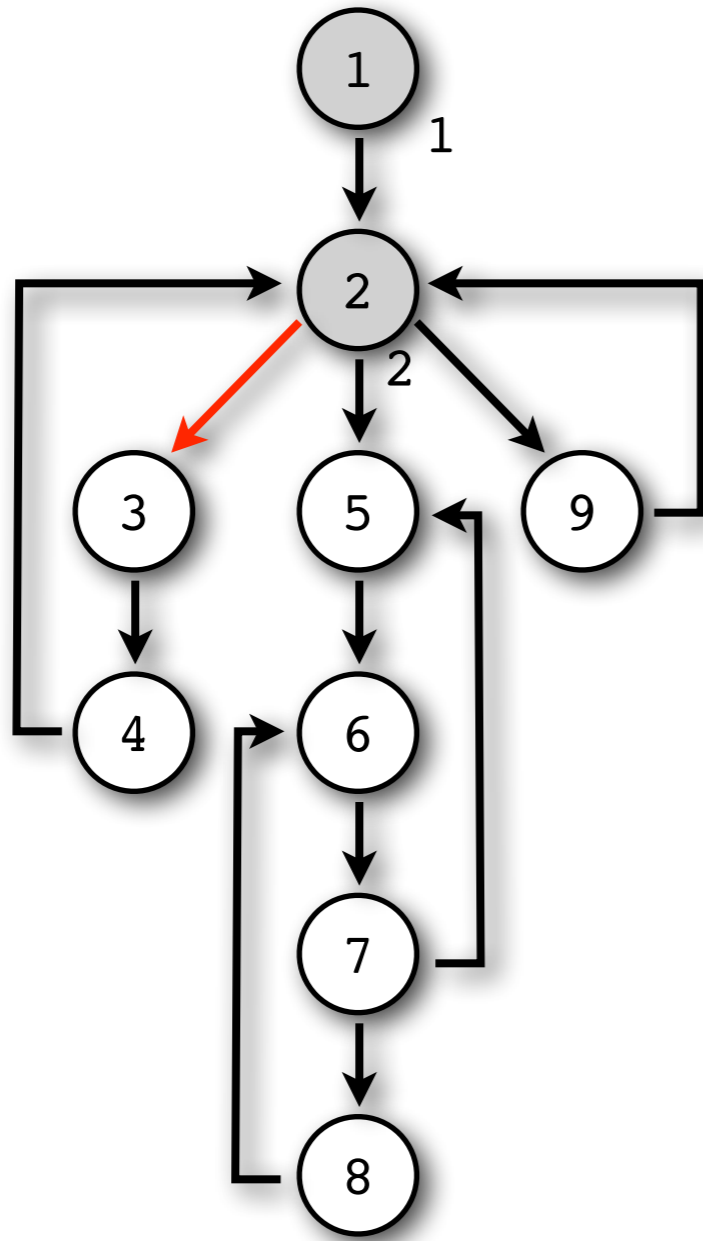
- on numérote les sommets par l'ordre de début de visite dans le parcours en profondeur
- on grise un sommet dont la visite a commencé
- on noircit un sommet dont la visite est terminée
- on utilise des lassos pour marquer les CFC calculées
- à chaque appel de $\text{Tarjan}(i)$, on indique sur le sommet i la valeur courante de min et on indique la valeur de la pile \mathcal{P} avant l'appel

Tarjan(1)

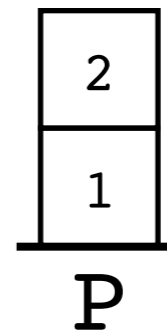
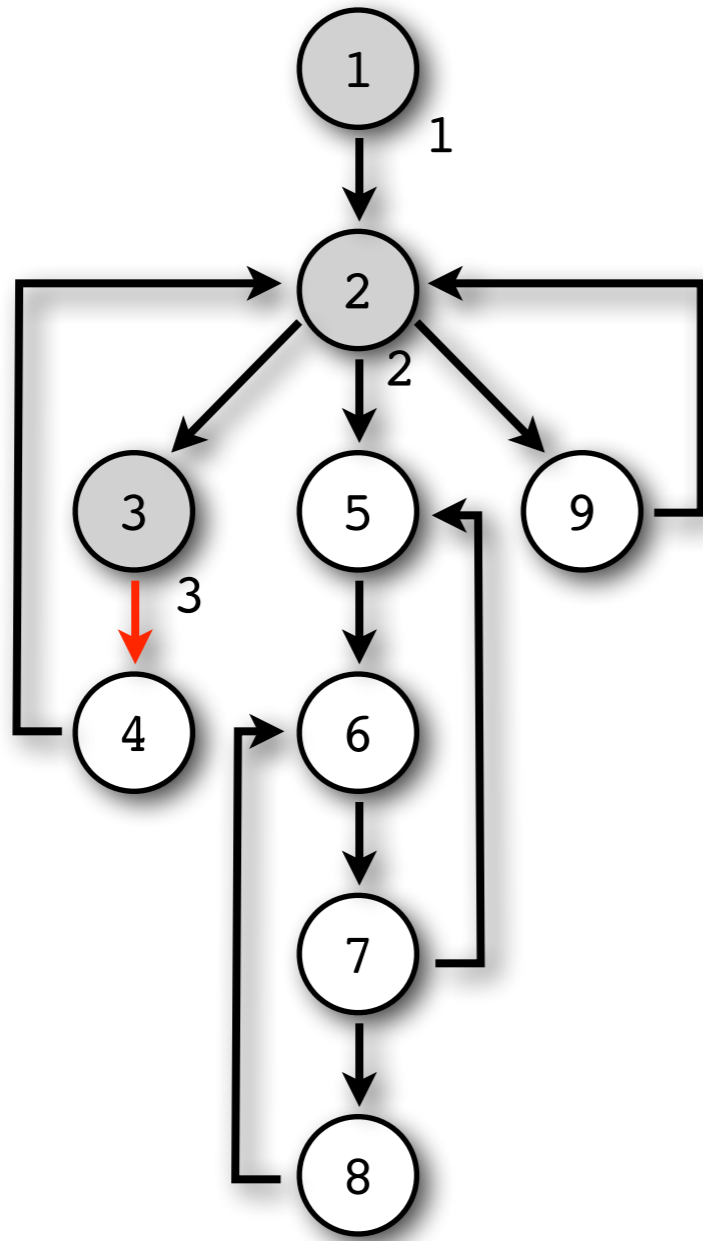


(vide)
P

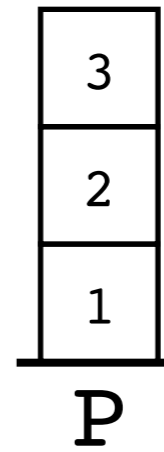
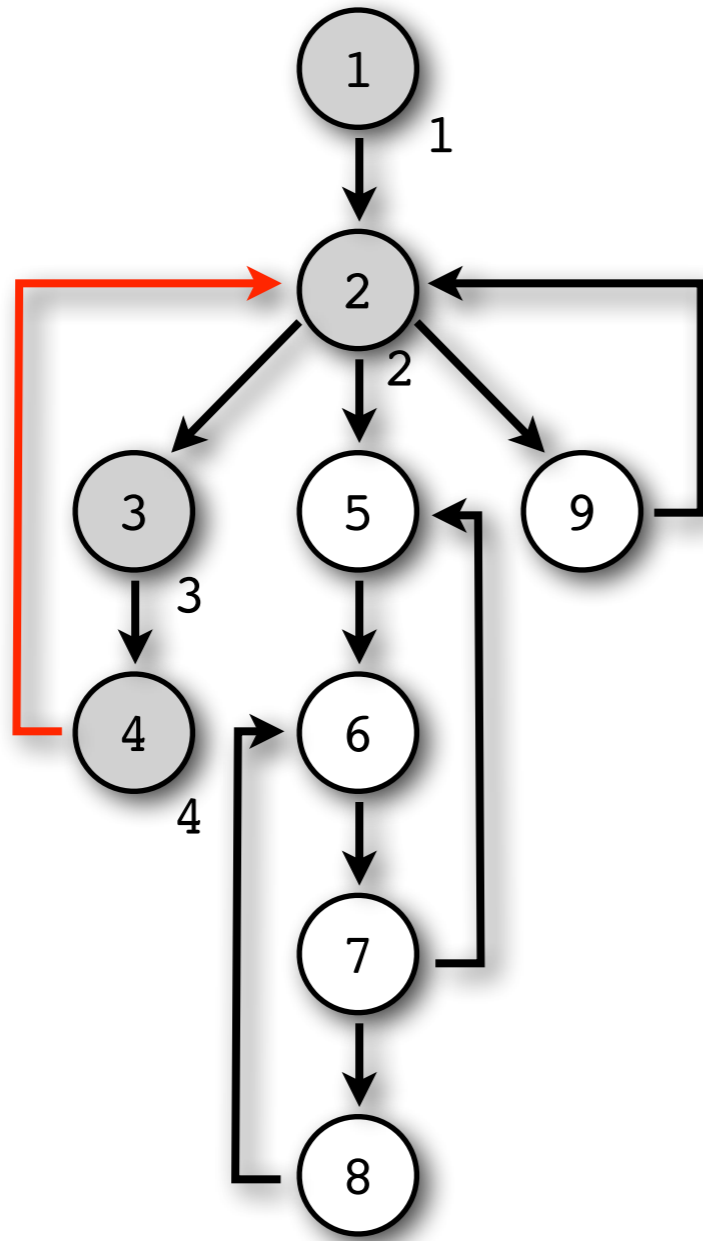
Tarjan(2)



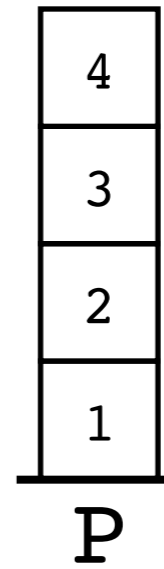
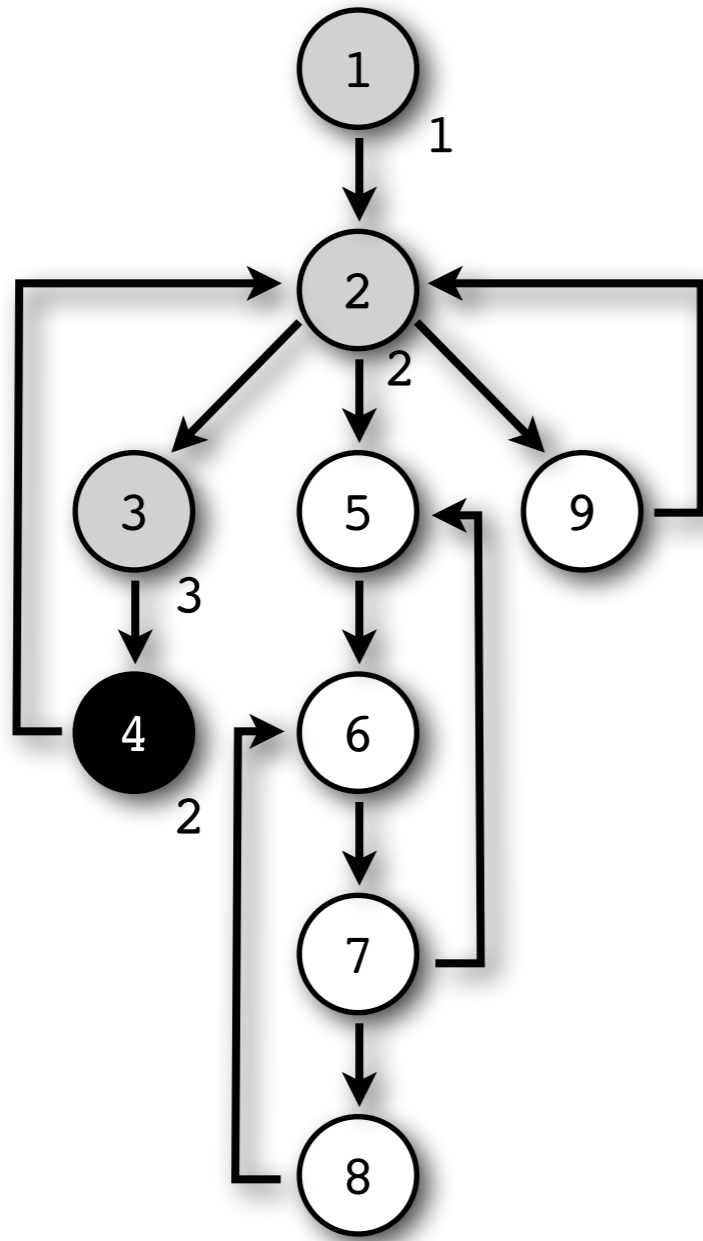
Tarjan(3)



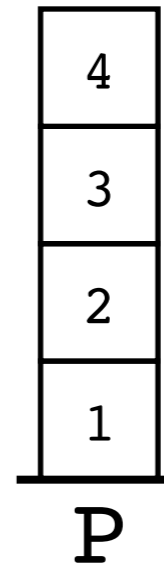
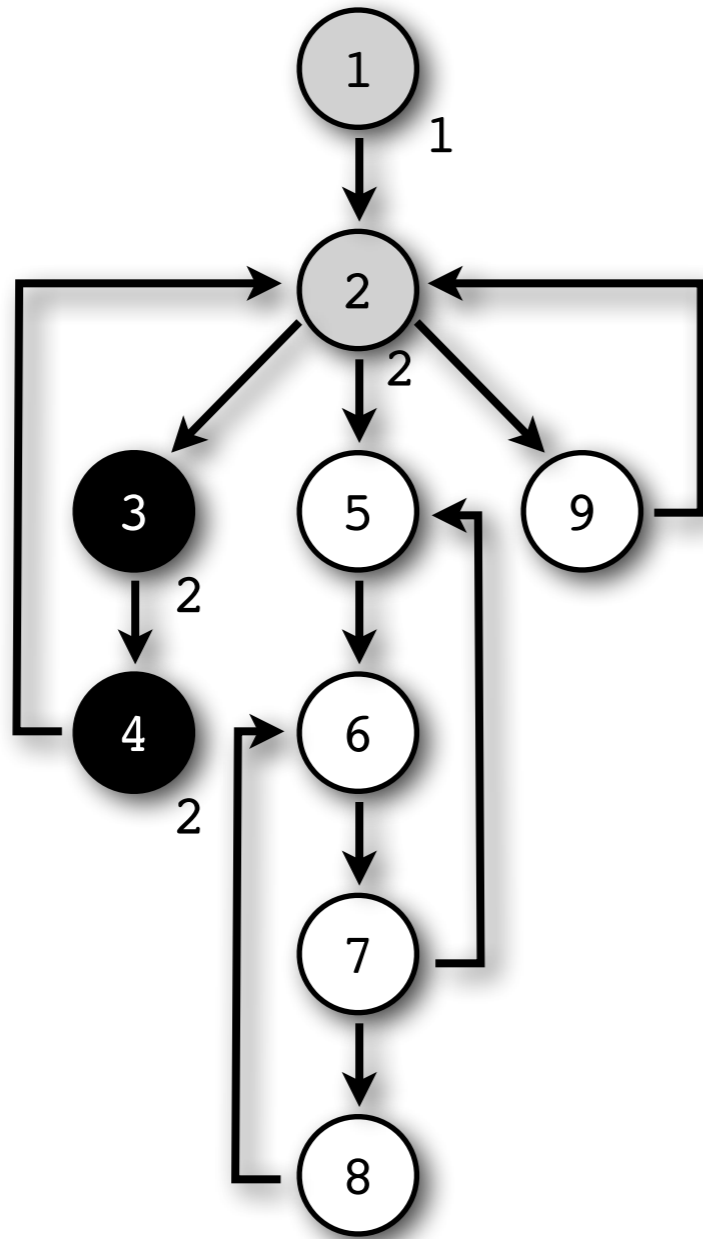
Tarjan(4)



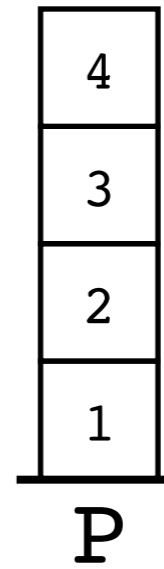
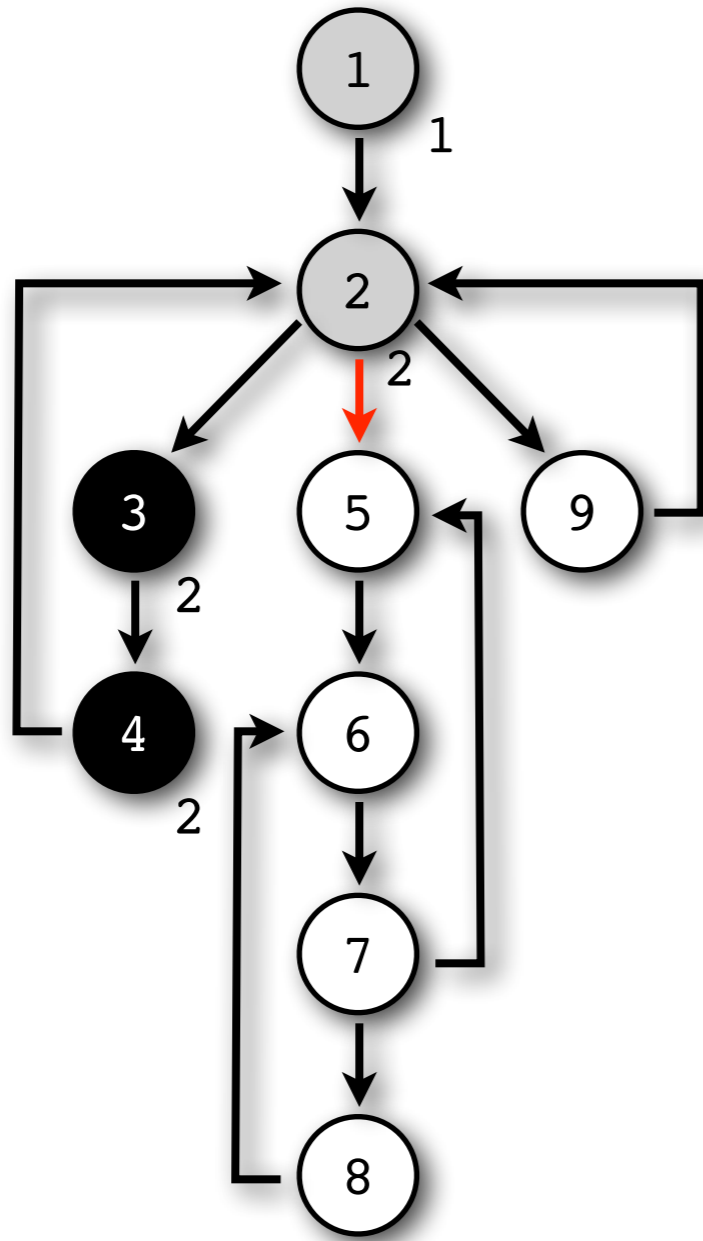
Tarjan(4)



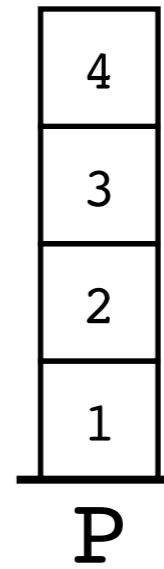
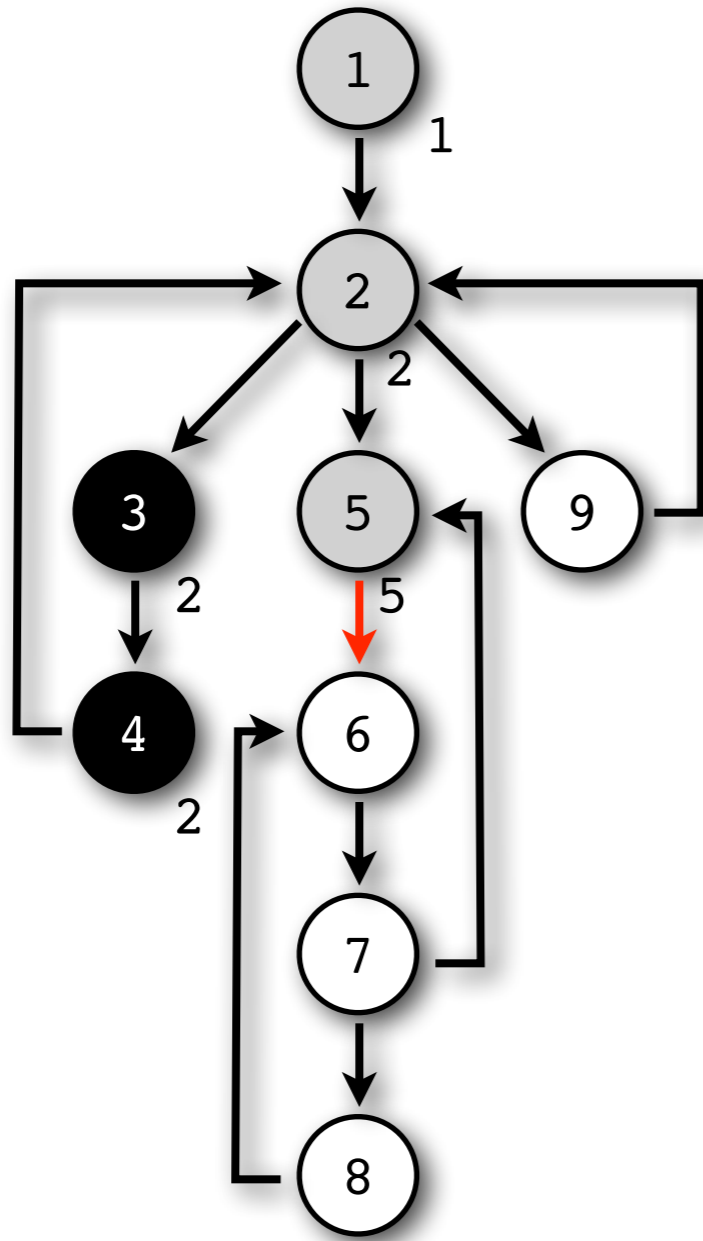
Tarjan(3)



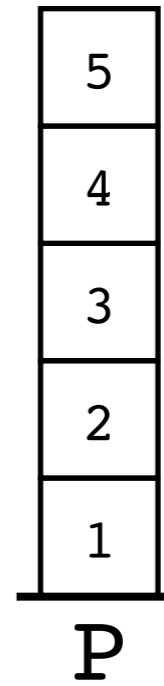
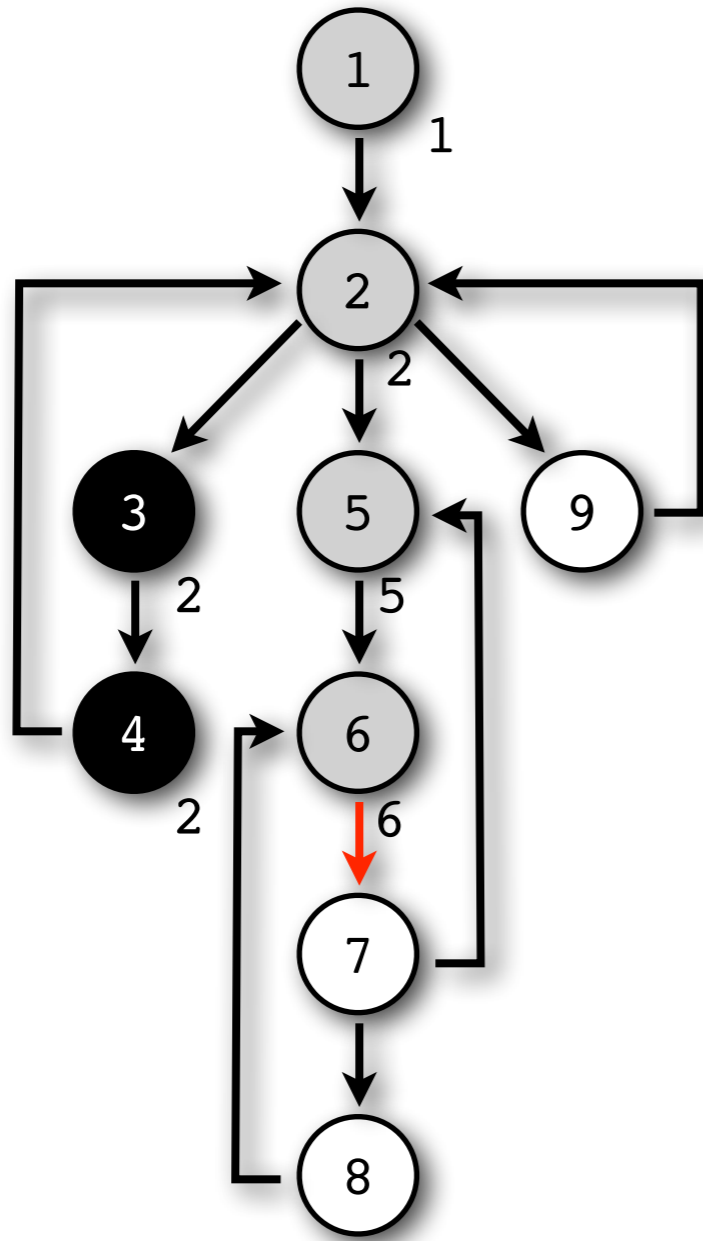
Tarjan(2)



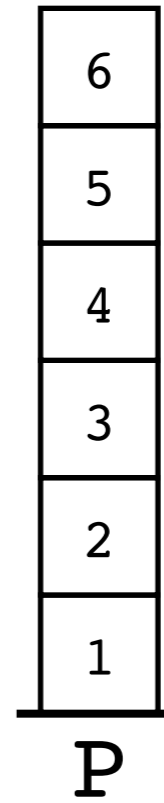
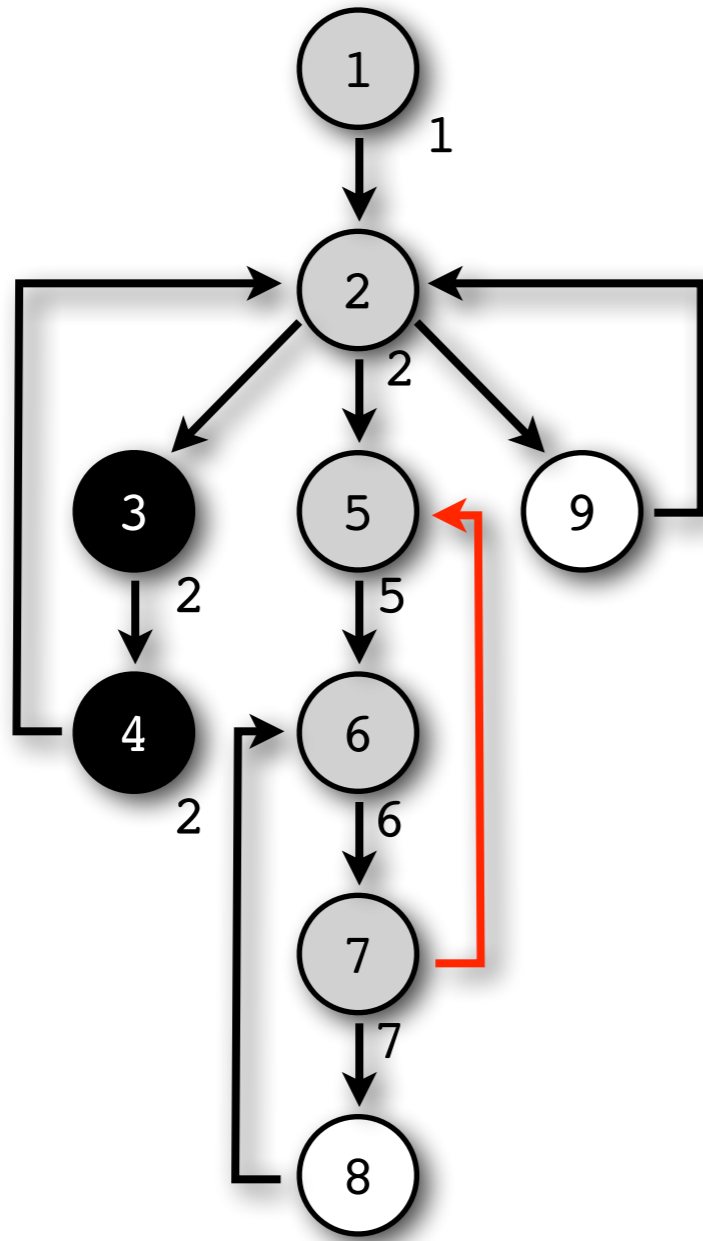
Tarjan(5)



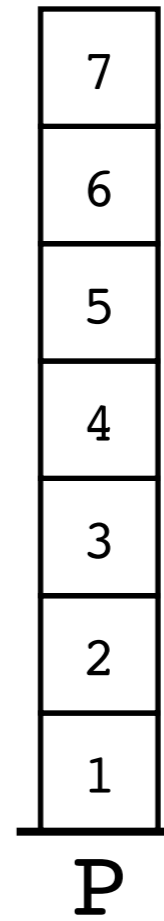
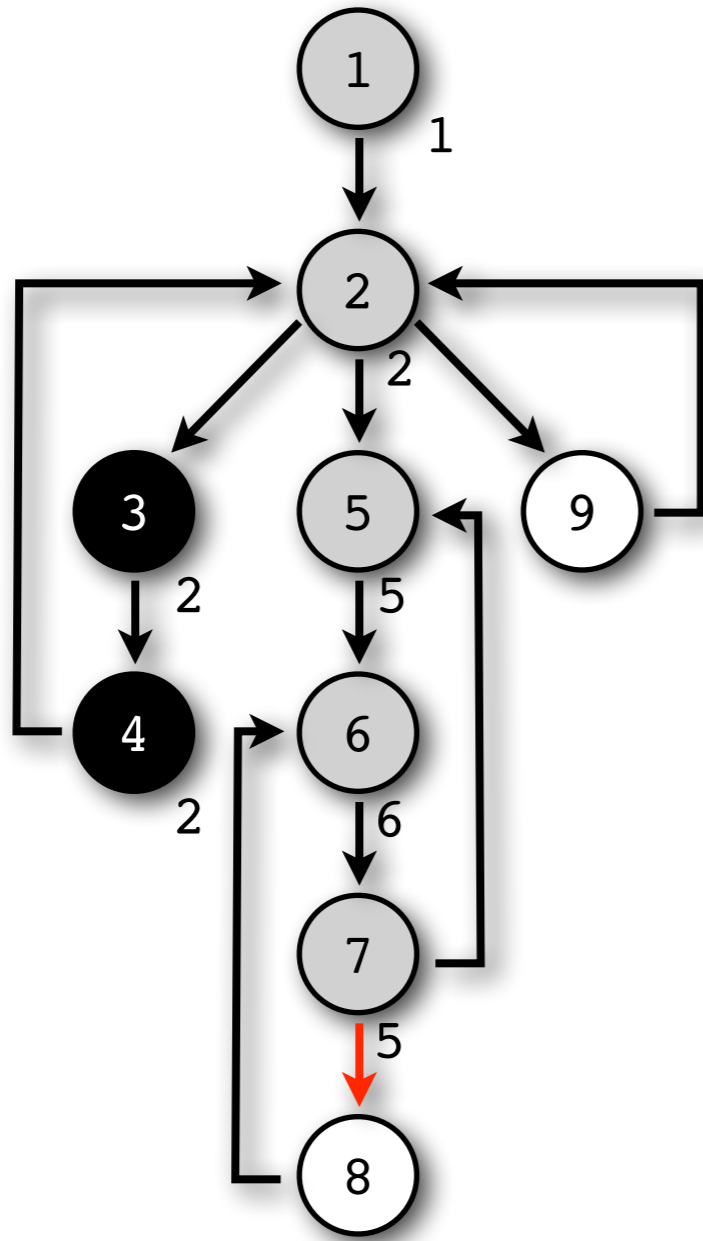
Tarjan(6)



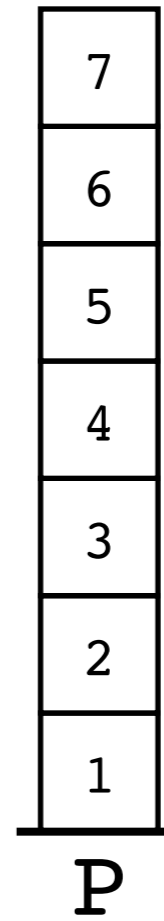
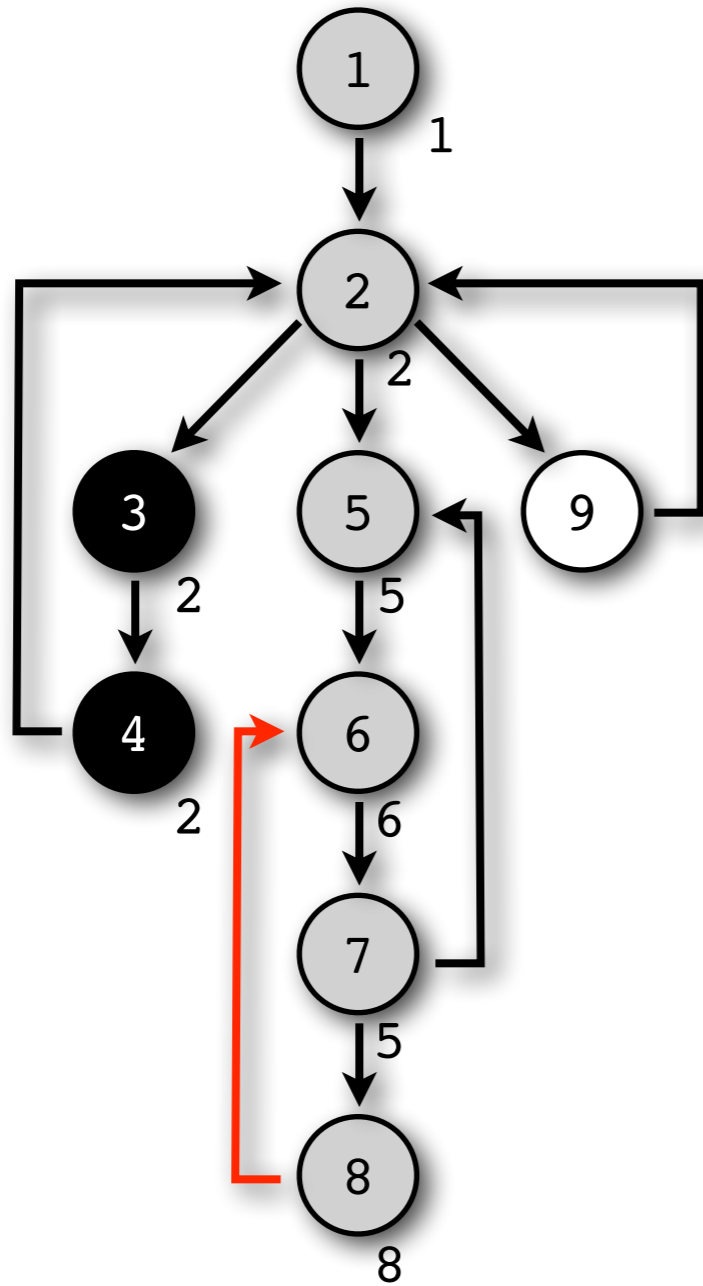
Tarjan(7)



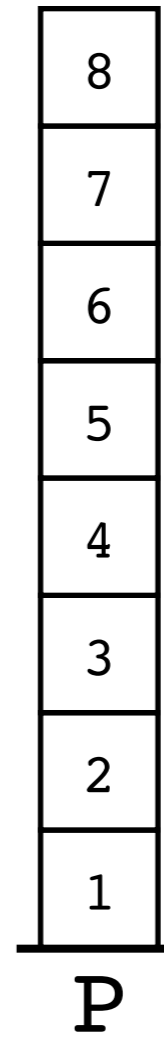
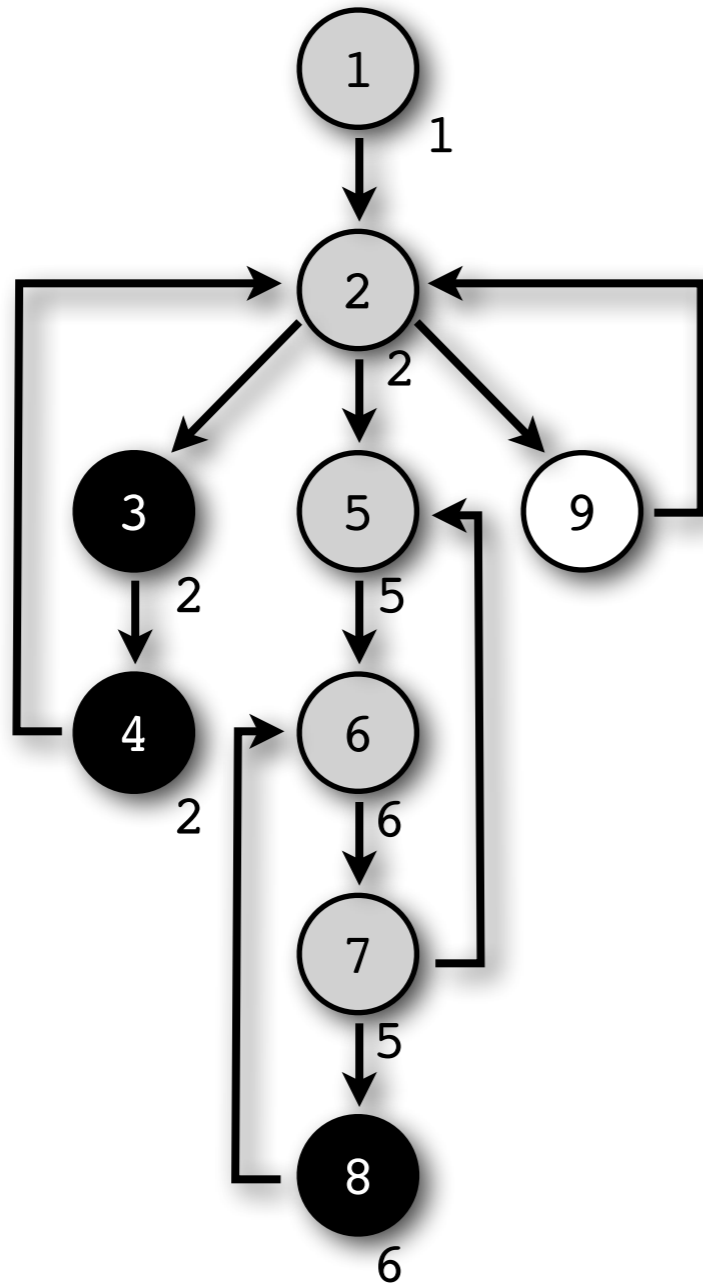
Tarjan(7)



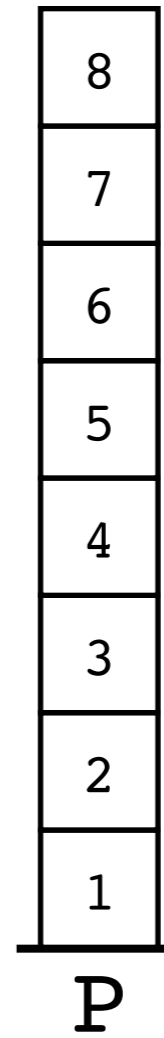
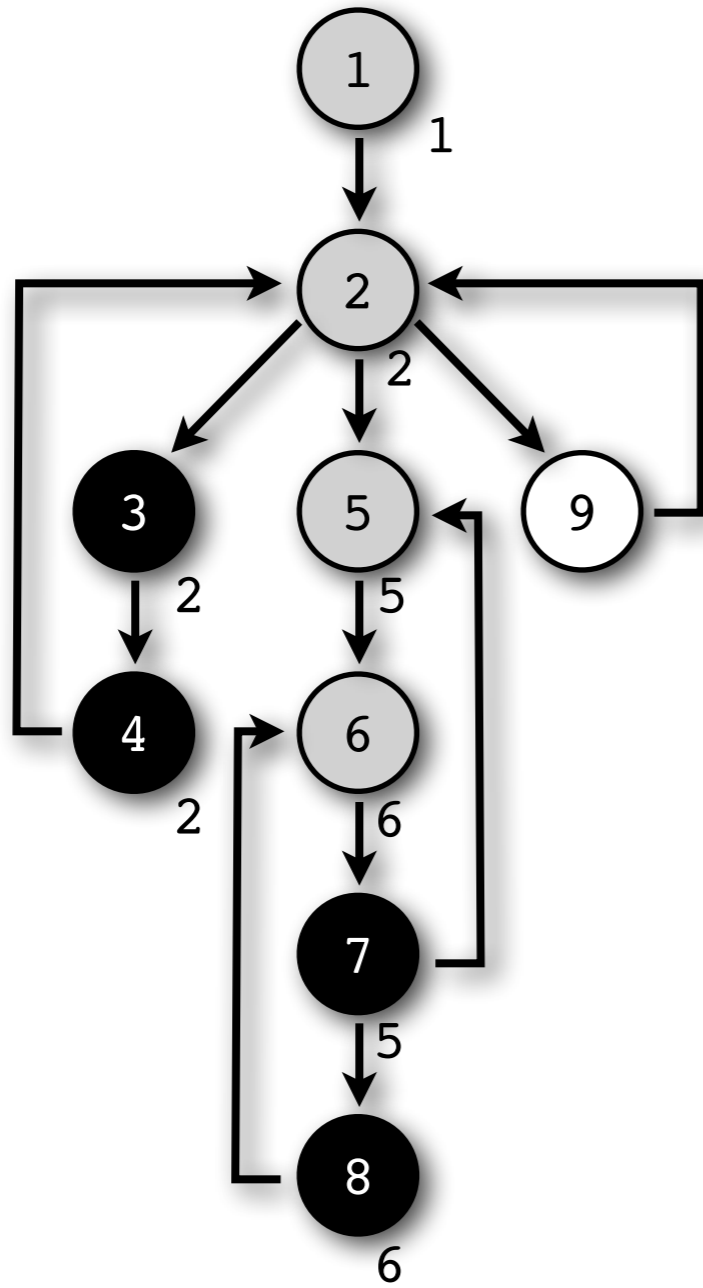
Tarjan(8)



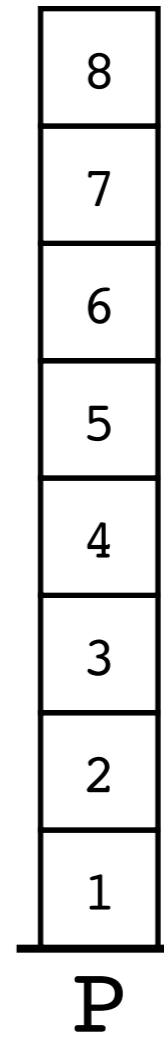
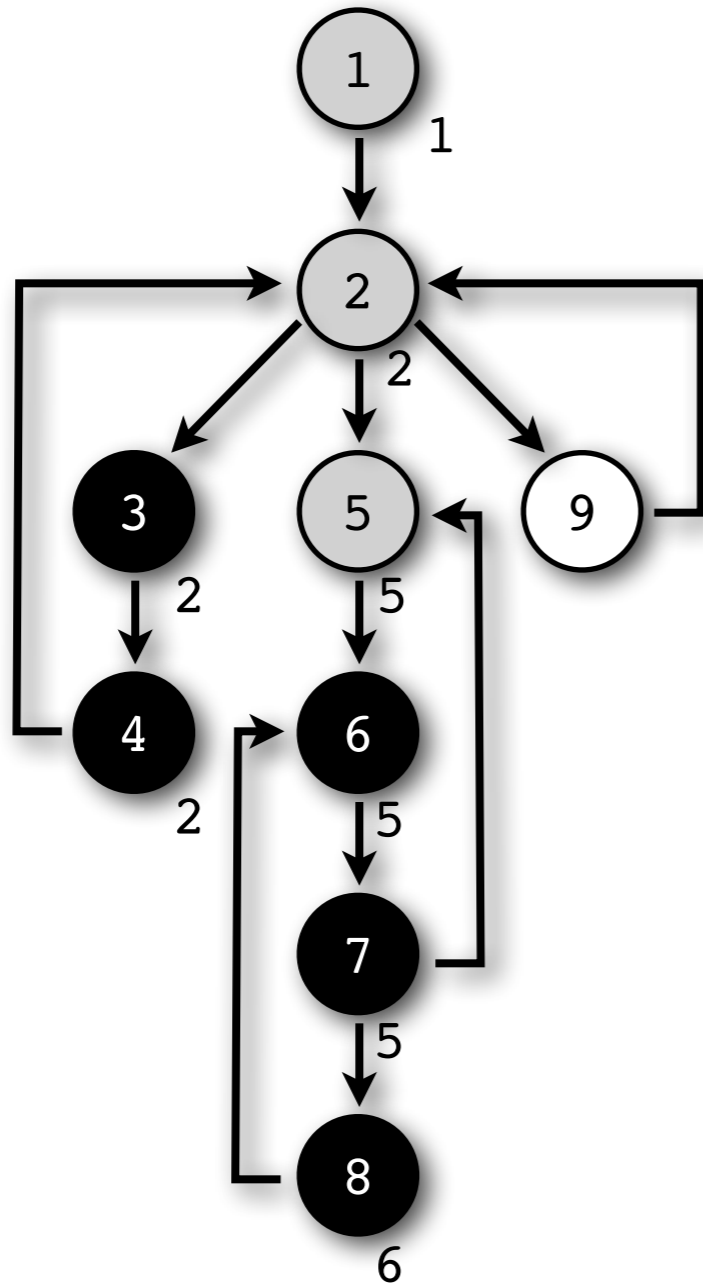
Tarjan(8)



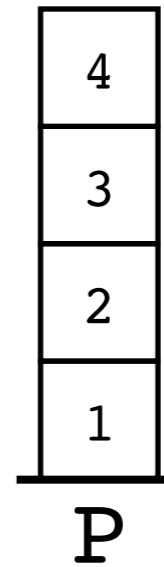
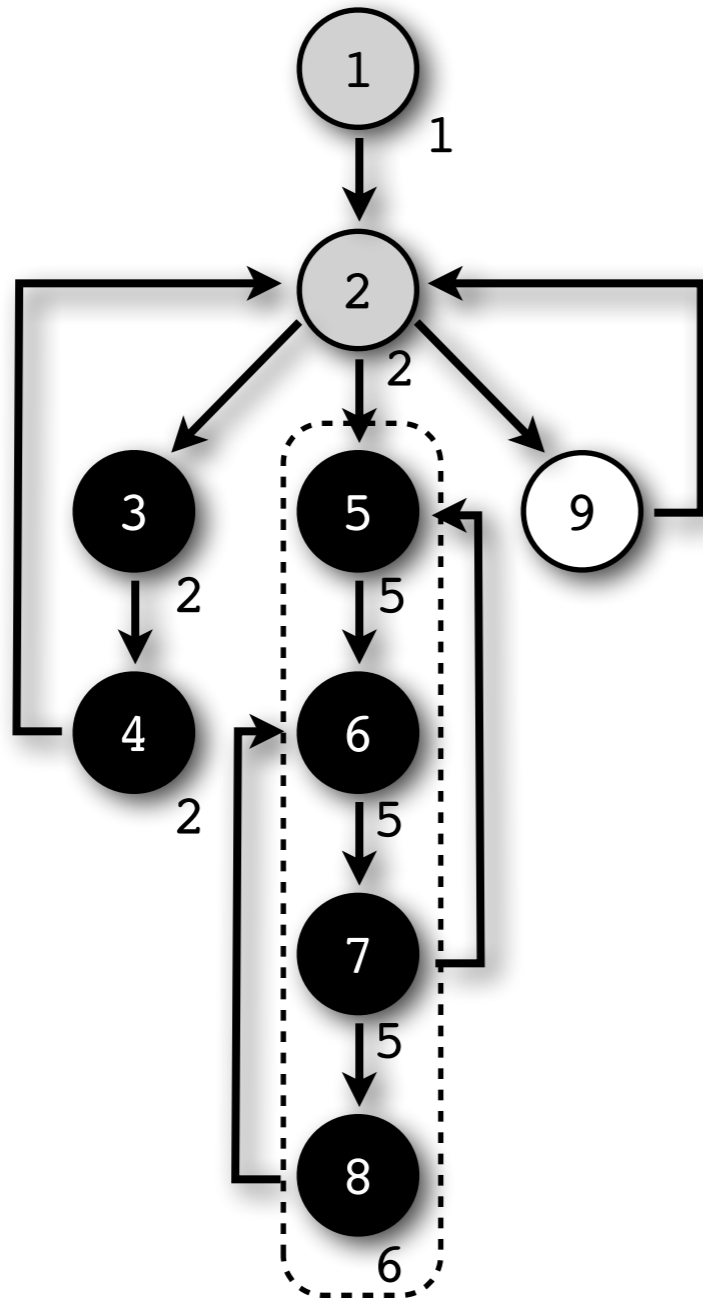
Tarjan(7)



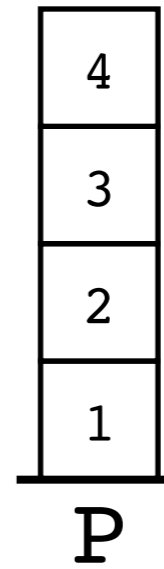
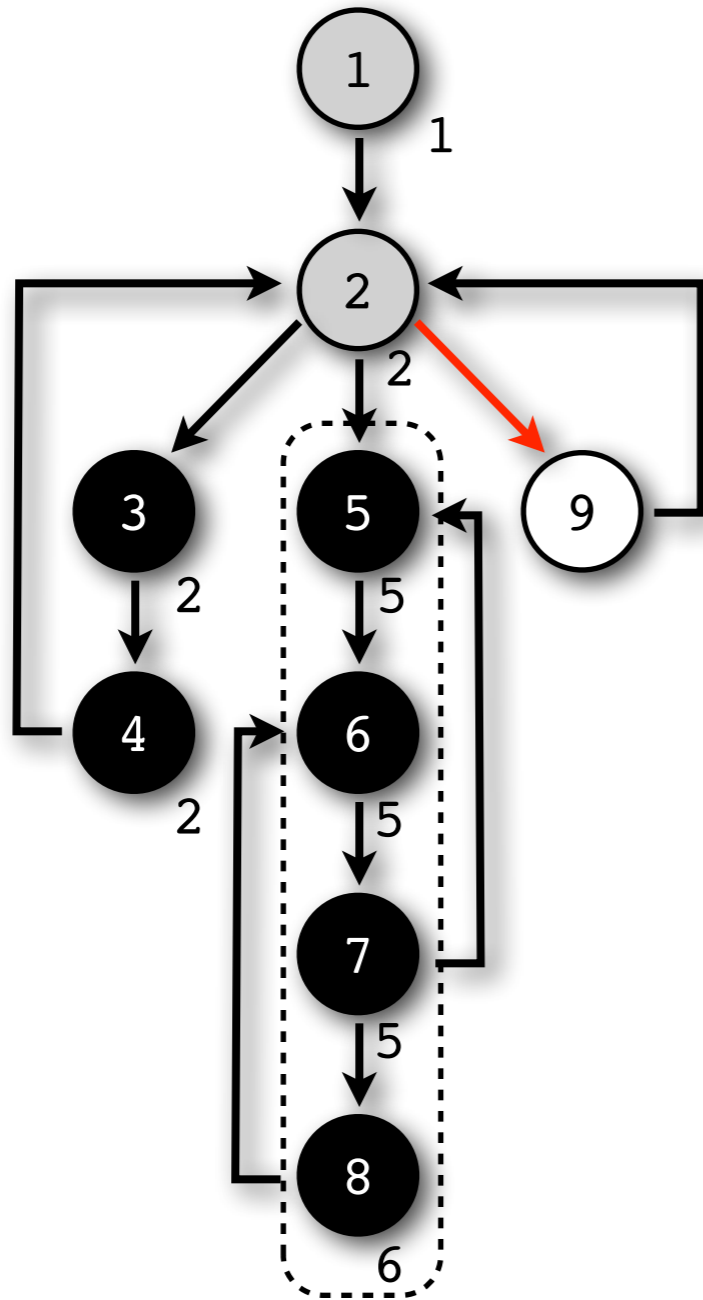
Tarjan(6)



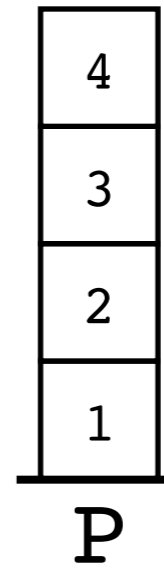
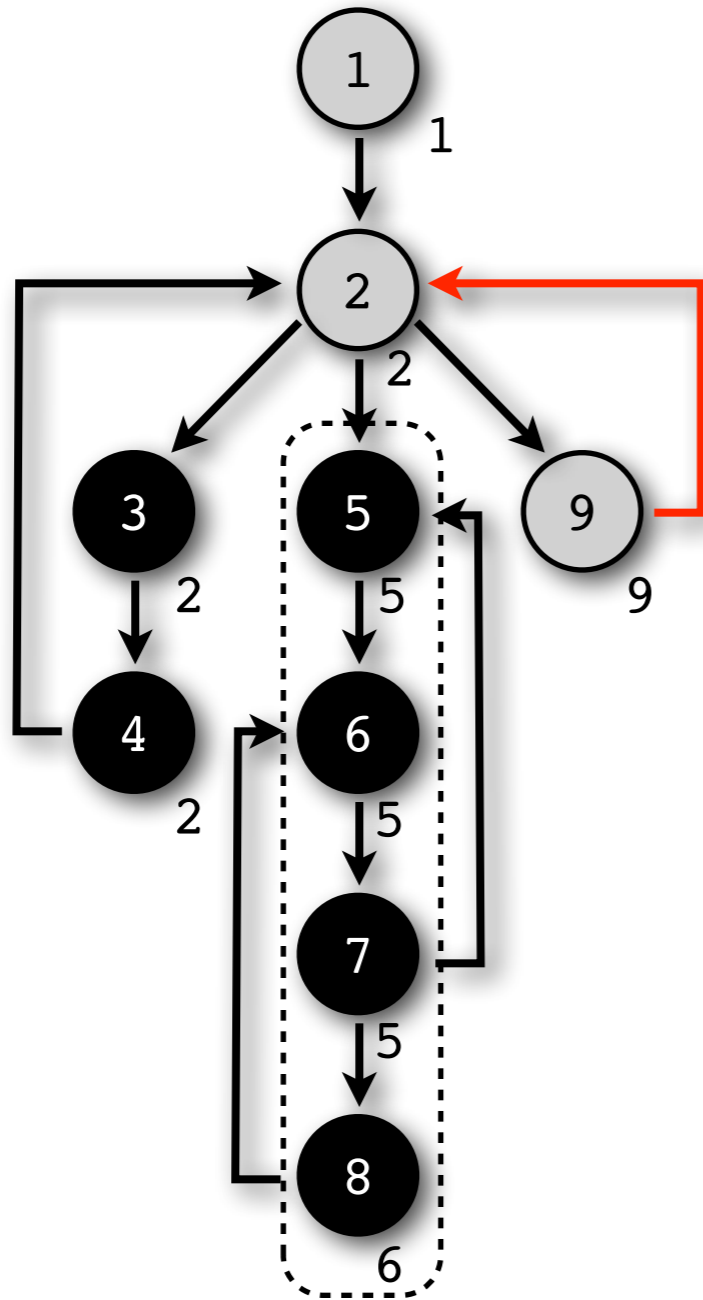
Tarjan(5)



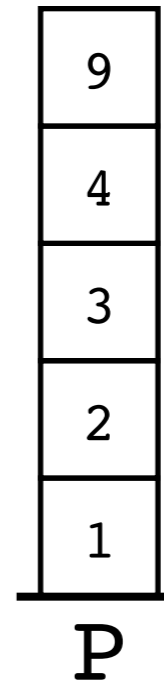
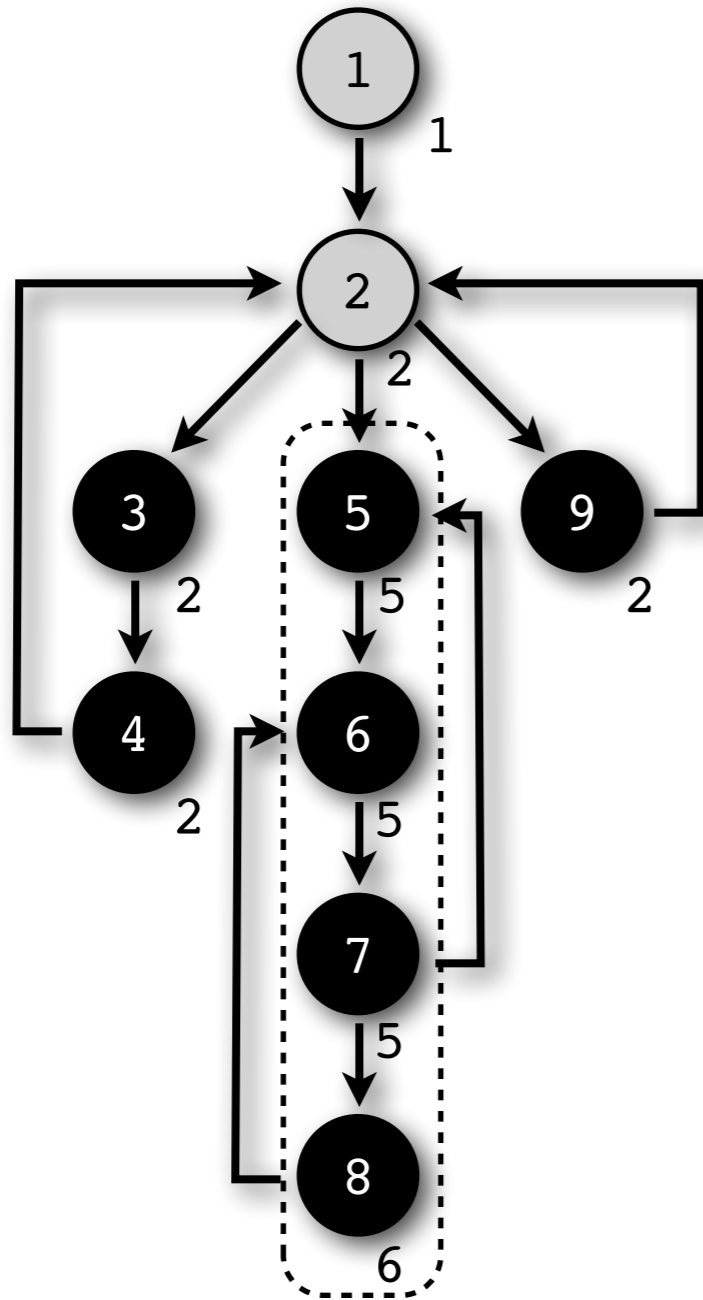
Tarjan(2)



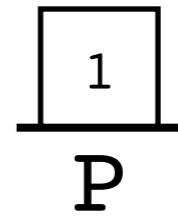
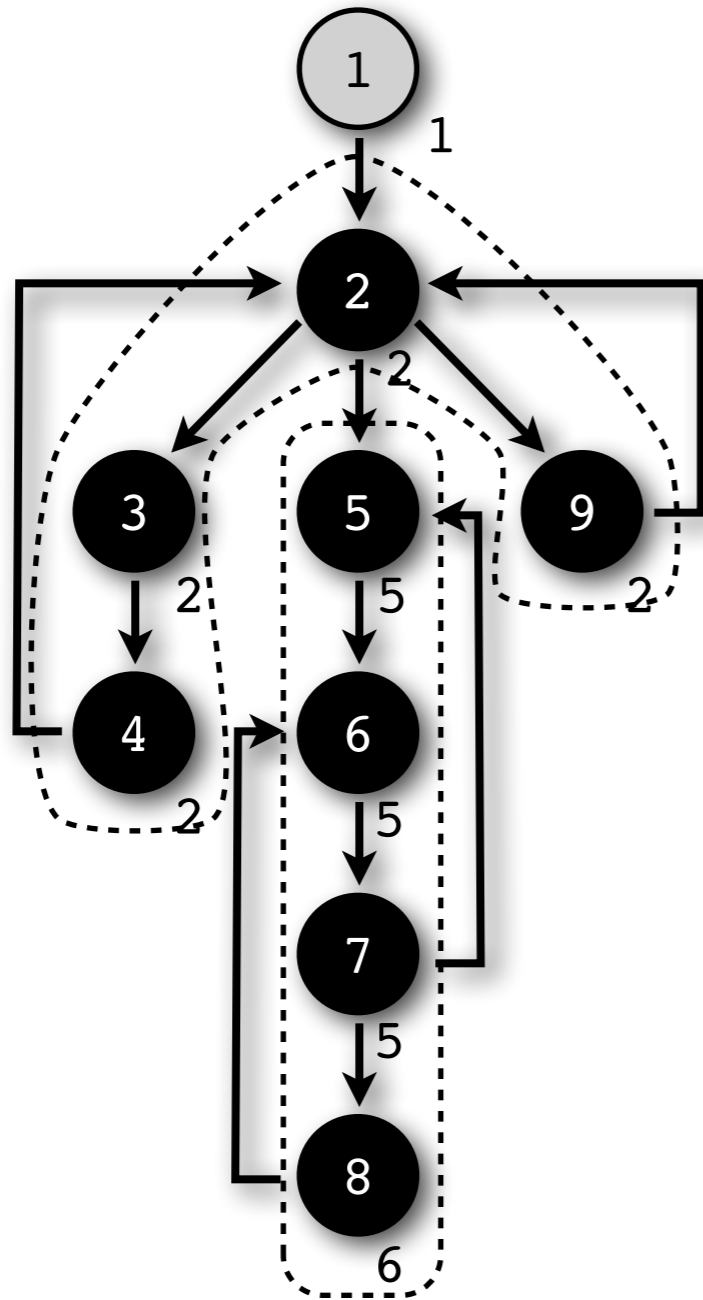
Tarjan(9)



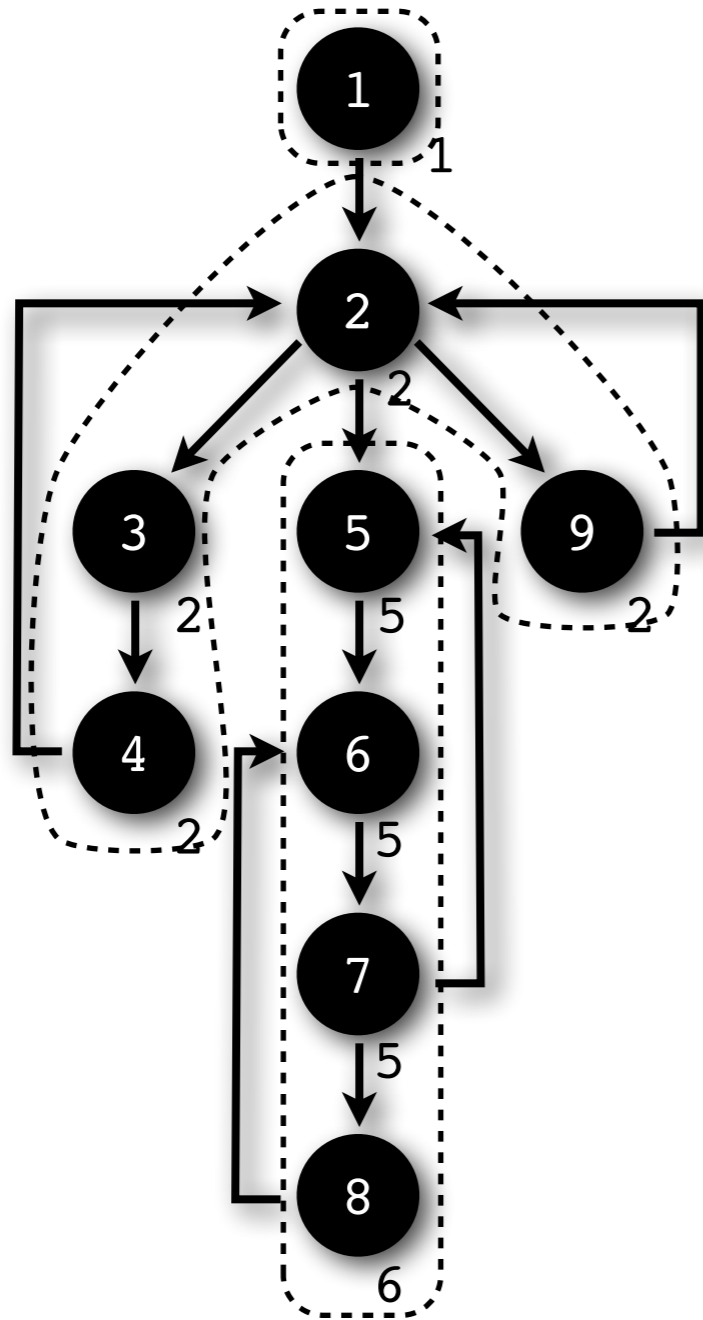
Tarjan(9)



Tarjan(2)



Tarjan(1)



(vide)
P