- 額外記錄 time stamp ⎧ discovery time : u.d
  ⎩ finishing time : u.f

- 和 BFS 不同 ⇒ 所有 vertex 皆會走訪

$\forall (u,v) \in E$

- edge 定義 state : ⓪ tree edge : v.color = white
  ① forward edge : v.color = black 且 u.d < v.d ⇒ u.d < v.d < v.f < u.f
  ② back edge : v.color = gray
  ③ cross edge : v.color = black 且 v.d < u.d ⇒ v.d < v.f < u.d < u.f

- 若 u 為 v 之 ancestor : $[v.d, v.f] \subseteq [u.d, u.f]$ ⇒ 判斷 ancestor 方式
  即: v.d < u.d < u.f < v.f

- 應用: ⓪ 找 connected component
  1. Initialization_graph (G)
  2. s = 0
  3. for each u ∈ G.V
     if u.color == white
     DFS_visit(G, u)
     s = s + 1

  在 DFS_visit(G, u) 中 將每個走訪點更新為 s 之值
  用 s 區別 component

  ① 判斷 G 是否含 cycle ⇒ 跑 DFS
  ⇒ DFS forest 中有無 back edge.