

```

// applicabile solo ai DAG, in quanto non hanno archi all'indietro
bool hasCycle(GRAPH  $G$ , NODE  $u$ , int &time, int[] dt, int[] ft)
    //  $u$ : il primo nodo che viene visitato

    time++ // aumento il contatore
    dt[ $u$ ] ← time // tempo di scoperta

    per ciascun  $u \in G.\text{adj}(u)$  fai
        se dt[ $v$ ] == 0 allora // non ho ancora scoperto questo nodo
            // effettuo una visita ricorsiva
            se hasCycle( $G, v$ ) allora
                └─ ritorna vero

        // logica dell'algoritmo
        altrimenti se dt[ $u$ ] > dt[ $v$ ] and ft[ $v$ ] == 0 allora
            // se raggiungo un mio discendente e non ho ancora terminato la mia visita,
            // allora ho trovato un arco all'indietro e quindi un ciclo
            └─ ritorna vero

    time++ // aumento il contatore
    ft[ $u$ ] ← time // tempo di fine

    // non ho trovato un ciclo
    ritorna falso

```