

```
// ricerca di un ciclo per grafi disconnessi
```

```
bool ciclico(GRAPH  $G$ )
```

```
    // inizializzazione
```

```
    bool[] visited  $\leftarrow$  new bool[1... $G.n$ ]
```

```
    per ciascun  $u \in G.V$  fai visited[ $u$ ]  $\leftarrow$  falso
```

```
    per ciascun  $u \in G.V$  fai
```

```
        se not visited[ $u$ ] allora
```

```
            // non ho ancora visitato il nodo
```

```
            se ciclicoRec( $G, v, \text{null}, \textit{visited}$ ) allora
```

```
                // effettuo una visita ricorsiva sul nodo vicino  $v$ 
```

```
                ritorna vero
```

```
    ritorna falso
```

```
// ricerca di un ciclo all'interno di un grafo
```

```
bool ciclicoRec(GRAPH  $G$ , NODE  $v$ , NODE  $p$ , bool[] visited)
```

```
    //  $p$ : nodo da cui provengo
```

```
    visited[ $u$ ]  $\leftarrow$  vero // visitato per la prima volta
```

```
    per ciascun  $v \in G.\text{adj}(u) - \{p\}$  fai
```

```
        //  $G.\text{adj}(u) - \{p\}$ : non considero  $(u, v)$  un ciclo nei grafi non orientati
```

```
        // per ogni nodo visito i suoi vicini
```

```
        se visited[ $v$ ] allora
```

```
            // se il nodo è già stato visitato allora ho trovato un ciclo
```

```
            ritorna vero
```

```
        altrimenti se ciclicoRec( $G, v, u, \textit{visited}$ ) allora
```

```
            // effettuo una visita ricorsiva sul nodo vicino  $v$ 
```

```
            ritorna vero
```

```
    ritorna falso
```