

## Visite di grafi

Dato un grafo  $G=(V,E)$ , gli algoritmi per la visita di un grafo partono da un vertice  $v \in V$  e visitano tutti i vertici che sono raggiungibili da  $v$  attraverso gli archi  $E$ :

Dato che un grafo può contenere dei cicli si utilizza un array booleano di appoggio *raggiunto[v]* tale per cui, dato  $v$ , *raggiunto[v] = TRUE* se il nodo è già stato scoperto.

### Visita BFS iterativa

Si utilizza una coda che inizialmente conterrà il nodo di partenza.

Finchè ho scoperto dei nodi non visitato per ogni vertice nella lista di adiacenza di  $u$  se non l'ho raggiunto allora setto *raggiunto[w]* a true e lo accodo.


Facendo questo visito il primo nodo, visito tutti i nodi adiacenti, ne prendo un altro, visito tutti i nodi adiacenti e così via.

#### Algoritmo visitaBFSIterativa(grafo $G=(V, E)$ , vertice $v$ )

- 1) *raggiunto[V]=FALSE*
- 2) coda  $C$
- 3) *enqueue(C,v)*
- 4) *raggiunto[v]=TRUE*
- 5) **while** ( $C \neq \text{NULL}$ )
  - a)  $u = \text{dequeue}(C)$
  - b) visita il vertice  $u$
  - c) **for each** vertex  $w$  of  $\text{Adj}[u]$   
    **if** ! *raggiunto[w]*
    - a) *raggiunto[w]=TRUE*
    - b) *enqueue(C,w)*

### Implementazione

Similmente agli alberi c'è un modulo per l'implementazione della coda.

<pre>struct elemBFS {     int inf;     elemBFS* pun ; };</pre>	 <p>Identificativo del nodo</p>	<pre>codaBFS enqueue(codaBFS, int); int dequeue(codaBFS&amp;); int first(codaBFS); static elemBFS* new_elem(int);</pre>
--	--	---

Grafo connesso: un grafo è connesso se al termine della visita *raggiunto[v] = TRUE* per ogni  $v$ .

### Componenti connesse

Una componente connessa è un sottografo  $G'$  di  $G$  connesso; ogni coppia di nodi di  $G'$  è connessa.

## **Albero BFS e albero di copertura**

Gli archi che conducono a vertici non visitati formano un albero detto albero BFS; la struttura dipende dall'ordine della visita.

Durante la visita possiamo registrare il padre di ogni nodo nel vettore dei padri in modo da risalire alla struttura.

Albero di copertura/spanning tree: albero BFS che include tutti i vertici di G.