

# Ricerca in un grafo - DFS

📅 Created	@January 21, 2025 4:14 PM
📖 Class	Prog2

## Ricerca in Profondità (DFS)

La **ricerca in profondità** (Depth-First Search, DFS) è un algoritmo di ricerca che esplora un grafo "in profondità", ossia partendo da un vertice, esplora ricorsivamente tutti i suoi vicini non visitati prima di passare al successivo. Il nome "profondità" deriva dal fatto che, in un albero, prima di visitare i nodi delle generazioni più vicine alla radice, l'algoritmo può ritrovarsi a visitare vertici lontani dalla radice stessa, andando così "in profondità".

### Caratteristiche principali:

- **Struttura dati:** Utilizza una **pila** (o la ricorsione) per tenere traccia dei vertici da visitare. Gli elementi vengono aggiunti alla pila in modo che l'algoritmo esplori sempre il vertice più profondo.
- **Complessità:** La complessità temporale è  $O(n + m)$ , dove  $n$  è il numero di vertici e  $m$  è il numero di archi, poiché ogni vertice viene visitato una volta e ogni arco viene esplorato al massimo una volta.
- **Uso:** DFS è utile per esplorazioni complete del grafo, per determinare se un grafo è connesso, o per risolvere problemi come il rilevamento di cicli.

```
DFS(G)
1. for each u in V[G]
2.     color[u]=white;
3.     pred[u]=NULL;
4. time = 0
5. for each u in V[G]
6.     if (color[u]==white)
7.         DFS-Visit(u)
```

```
DFS-Visit(u)
1. color[u]=grey;
  d[u]=time++;
2. for each v in Adj[u]
3.     if (color[v]==white)
4.         pred[v]=u;
5.         DFS-Visit(v);
6. color[u]=black
7. f[u]=time++;
```

### **Procedimento sintetizzato:**

1. **Visitare il nodo iniziale** e marcarlo come visitato.
2. **Esplorare ricorsivamente i vicini non visitati** di quel nodo, scegliendo uno alla volta.
3. **Tornare indietro** nel momento in cui non ci sono più vicini non visitati, esplorando quindi il nodo precedente nella pila.
4. Ripetere il processo fino a che tutti i nodi sono stati visitati.

**Nota sulla Foresta DFS:** Il sottografo dei predecessori generato dalla visita in profondità può essere costituito da più alberi. Questo sottografo è definito **foresta DFS**, composta da diversi alberi DFS, dove ogni albero rappresenta una componente connessa del grafo.

### **Differenze rispetto a BFS:**

- **Comportamento:** DFS esplora "in profondità", andando il più lontano possibile lungo ogni ramo prima di tornare indietro. Al contrario, BFS esplora il grafo "orizzontalmente", visitando tutti i vicini di un nodo prima di passare a quelli di livello superiore.
- **Struttura dati:** DFS utilizza una pila (LIFO) o la ricorsione, mentre BFS usa una coda (FIFO). In DFS, i vertici vengono esplorati in ordine inverso rispetto a quello in cui sono stati aggiunti, dando priorità ai nodi più "profondi".

### **Confronto con BFS:**

- **DFS** è preferibile quando si vuole esplorare tutto il grafo e si ha bisogno di una visita completa. Inoltre, è utile per problemi come la verifica di cicli e la ricerca di componenti connesse.
- **BFS**, invece, è spesso utilizzato per trovare il cammino più breve in un grafo non pesato, mentre DFS non è progettato per questo scopo.