

Duyệt theo chiều sâu trên đồ thị: Depth-First Search (DFS)

- Thực hiện một thủ tục thăm các đỉnh của đồ thị (mỗi đỉnh đúng 1 lần) dọc theo các cạnh (cung).

Duyệt theo chiều sâu trên đồ thị: Depth-First Search (DFS)

- Thực hiện một thủ tục thăm các đỉnh của đồ thị (mỗi đỉnh đúng 1 lần) dọc theo các cạnh (cung).
- $\text{DFS}(u)$: duyệt theo chiều sâu bắt đầu từ đỉnh u
 - Tiến hành thăm u

Duyệt theo chiều sâu trên đồ thị: Depth-First Search (DFS)

- Thực hiện một thủ tục thăm các đỉnh của đồ thị (mỗi đỉnh đúng 1 lần) dọc theo các cạnh (cung).
- $\text{DFS}(u)$: duyệt theo chiều sâu bắt đầu từ đỉnh u
 - Tiến hành thăm u
 - Nếu tồn tại đỉnh v trong danh sách kề của u chưa được thăm thì gọi đệ quy $\text{DFS}(v)$ để duyệt theo chiều sâu bắt đầu từ v

Duyệt theo chiều sâu trên đồ thị: Depth-First Search (DFS)

- Thực hiện một thủ tục thăm các đỉnh của đồ thị (mỗi đỉnh đúng 1 lần) dọc theo các cạnh (cung).
- $DFS(u)$: duyệt theo chiều sâu bắt đầu từ đỉnh u
 - Tiến hành thăm u
 - Nếu tồn tại đỉnh v trong danh sách kề của u chưa được thăm thì gọi đệ quy $DFS(v)$ để duyệt theo chiều sâu bắt đầu từ v
 - Nếu tất cả các đỉnh kề với u đã được thăm thì DFS quay trở lại đỉnh x mà từ đó thăm u để tiến hành thăm các đỉnh khác kề với x mà chưa được thăm. Lúc này đỉnh u được gọi là *đã duyệt xong*

Duyệt theo chiều sâu trên đồ thị: Depth-First Search (DFS)

- Thực hiện một thủ tục thăm các đỉnh của đồ thị (mỗi đỉnh đúng 1 lần) dọc theo các cạnh (cung).
- $DFS(u)$: duyệt theo chiều sâu bắt đầu từ đỉnh u
 - Tiến hành thăm u
 - Nếu tồn tại đỉnh v trong danh sách kề của u chưa được thăm thì gọi đệ quy $DFS(v)$ để duyệt theo chiều sâu bắt đầu từ v
 - Nếu tất cả các đỉnh kề với u đã được thăm thì DFS quay trở lại đỉnh x mà từ đó thăm u để tiến hành thăm các đỉnh khác kề với x mà chưa được thăm. Lúc này đỉnh u được gọi là *đã duyệt xong*

- Xét đồ thị $G = (V, A)$ trong đó
 - V : tập đỉnh
 - $A(v)$: tập các đỉnh kề với v
- Cấu trúc đánh dấu
 - $visited[v] = \text{true}$: đỉnh v đã được thăm

```
DFS(u, A) {  
    Thăm đỉnh u;  
    visited[u] = true;  
    for v ∈ A(u) do {  
        if(visited[v] = false) {  
            DFS(v, A);  
        }  
    }  
}
```

```
DFS(V, A) {  
    for v ∈ V do {  
        visited[v] = false;  
    }  
    for v ∈ V do {  
        if(visited[v] = false) {  
            DFS(v, A);  
        }  
    }  
}
```