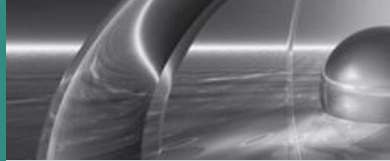


Tìm kiếm trên đồ thị theo chiều sâu

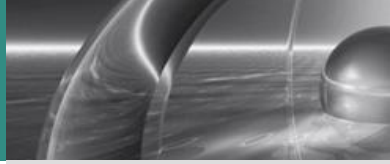
Cài đặt bằng Độ quy



```
void DFS(int s) {  
    if (visited[s] == true) return;  
    // Bước 1  
    visited[s] = true;  
  
    // process node s:..  
  
    // Bước 2  
    foreach (int u in v[s])  
        DFS(u);  
}
```

Tìm kiếm trên đồ thị theo chiều sâu

Cài đặt bằng Stack



```
// Dùng stack
void DFS(int s)
{
    1. Đánh dấu s đã viếng thăm
    2. Đưa s vào stack
    3. while stack chưa rỗng
    {
        <Lấy 1 đỉnh u từ stack>
        <Tìm 1 đỉnh v kề u và chưa được viếng thăm>
        {
            Đánh dấu v đã viếng thăm
            Đẩy u vào lại stack
            Đẩy v vào stack
        }
    }
}
```

Tìm kiếm trên đồ thị theo chiều rộng

Cài đặt bằng hàng đợi

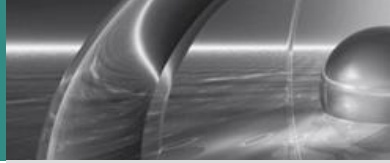
```
void BFS(int s)
{
    visited[s]=true;
    q.push(s);
    // Process node s

    while (q.Count!=0) {
        s = q.front();
        q.pop()

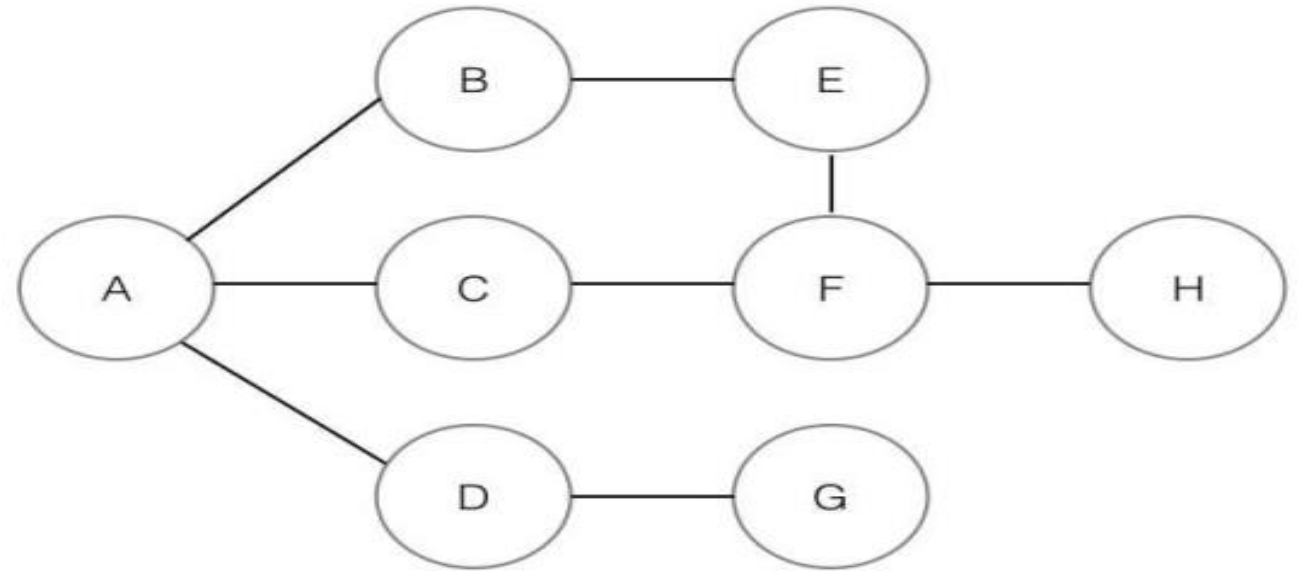
        foreach (int u in v[s]) {
            if (visited[u]) continue;
            visited[u]=true;
            q.push(u);

            // Process node u
        }
    }
}
```

DFS & BFS (từ A)



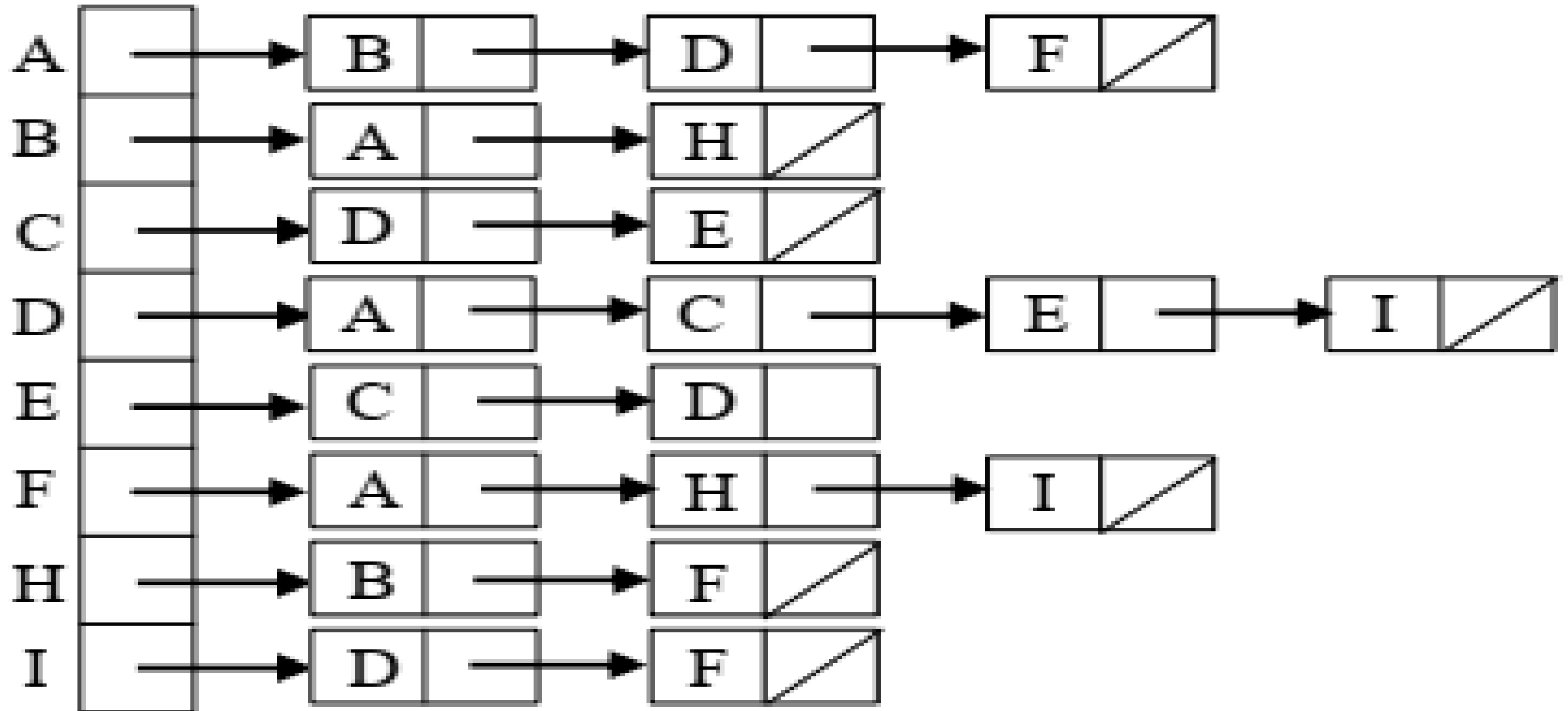
	A	B	C	D	E	F	G	H
A	0	1	1	1	0	0	0	0
B	1	0	0	0	1	0	0	0
C	1	0	0	0	0	1	0	0
D	1	0	0	0	0	0	1	0
E	0	1	0	0	0	1	0	0
F	0	0	1	0	1	0	0	1
G	0	0	0	1	0	0	0	0
H	0	0	0	0	0	1	0	0



A: B, C, D
B: A, E
C: A, F
D: A, G
E: B, F
F: C, E, H
G: D
H: F

DFS (stack)	BFS (queue)
$S = [A]$	$Q = [A]$
$S = [D, C, B]$	$Q = [B, C, D]$
$S = [G, C, B]$	$Q = [C, D, E]$
$S = [C, B]$	$Q = [D, E, F]$
$S = [F, B]$	$Q = [E, F, G]$
$S = [H, E, B]$	$Q = [F, G]$
$S = [E, B]$	$Q = [G, H]$
$S = [B]$	$Q = [H]$
$S = []$	$Q = []$

Ví dụ 1 (tự làm): DFS & BFS (từ I)



Ví dụ 2 (tự làm): DFS & BFS (từ A)

