

GSI011 – Estrutura de Dados 2

Prof^a: Christiane Regina Soares Brasil

GSI011 – Estrutura de Dados 2 – Aula 8

- Busca em grafo:
 - É um processo específico de como percorrer por seus vértices e arestas.
 - Dependem do vértice inicial (ponto de partida).
 - Tipos de busca em grafos:
 - Busca em Largura
 - Busca em Profundidade

GSI011 – Estrutura de Dados 2 – Aula 8

- Busca em Largura:
 - Inicia-se em um vértice v_i e então visita todos os seus vizinhos, antes de se aprofundar na busca. Esse processo continua até que:
 - O alvo seja encontrado
 - Não existam mais vértices a serem visitados
 - Para realizar essa tarefa é utilizada uma fila FIFO (**First In, First Out**).

GSI011 – Estrutura de Dados 2 – Aula 8

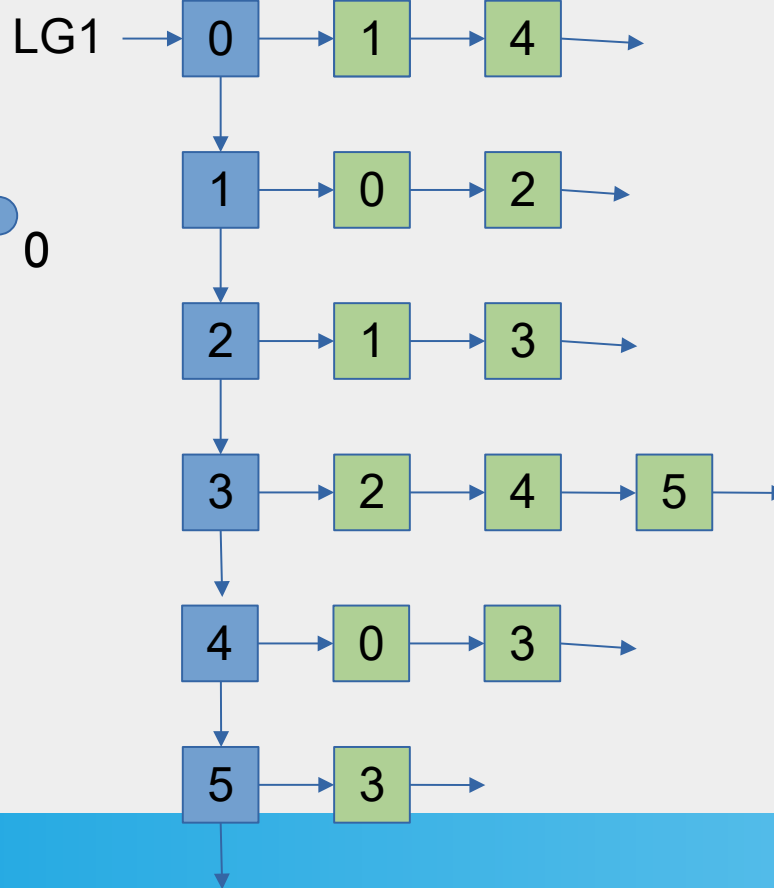
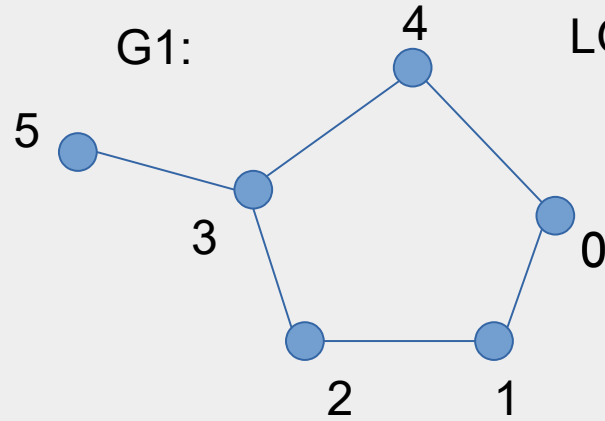
- **Busca em Largura:**

Vamos supor um grafo G com n vértices e vi sendo o ponto de partida, que está representado na lista de adjacência Adj .

Complexidade: $O(n + m)$

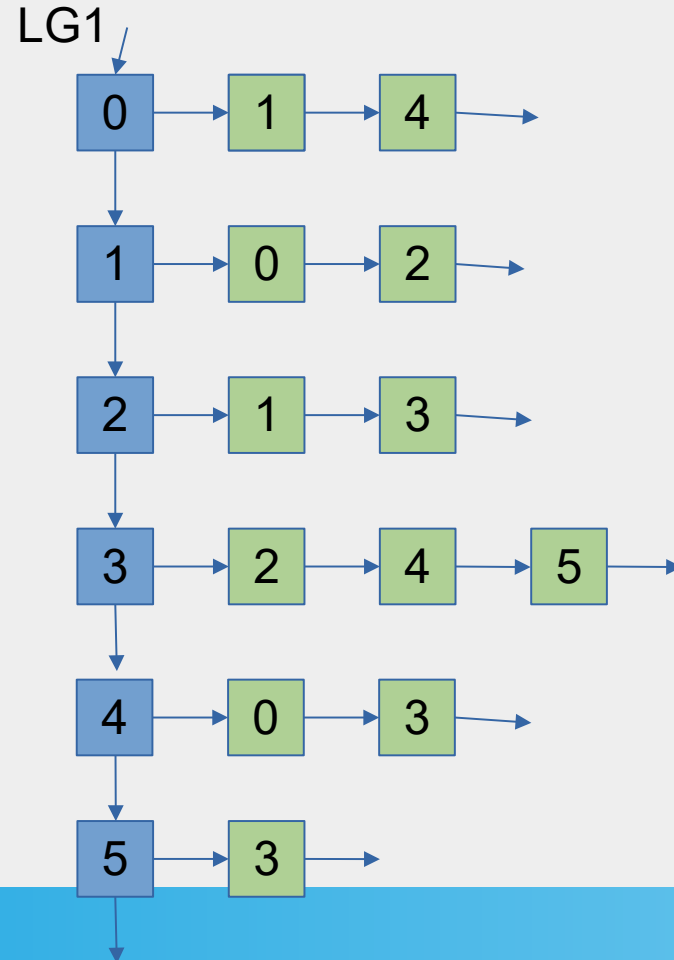
```
BuscaLargura(vi, n, Adj)
INICIO
cont ← 0;
para u ← 0 até n – 1 faça visitado[u] ← 0;
visitado[vi] = cont;    F ← criaFila(vi);
enquanto (F não vazia) faça
    v ← saiFila(F);
    enquanto houver w em Adj[v] faça
        se (visitado[w] == 0) então
            cont++;
            visitado[w] ← cont;
            entraFila(F,w);
        fimse
    fimenquanto
    liberaFila(F);
retorna visitado[0...n-1];
FIM
```

GSI011 – Estrutura de Dados 2 – Aula 8



GSI011 – Estrutura de Dados 2 – Aula 8

```
BuscaLargura(vi, n, Adj)
INICIO
cont ← 1; // ordem da visitação
para u ← 0 até n – 1 faça visitado[u] ← 0;
visitado[vi] = cont; F ← criaFila(vi);
enquanto (F não vazia) faça
    v ← saiFila(F);
    enquanto houver w em Adj[v] faça
        se (visitado[w] == 0) então
            cont++;
            visitado[w] ← cont;
            entraFila(F, w);
    fimse
    fimenquanto
    liberaFila(F);
retorna visitado[0...n-1];
FIM
```

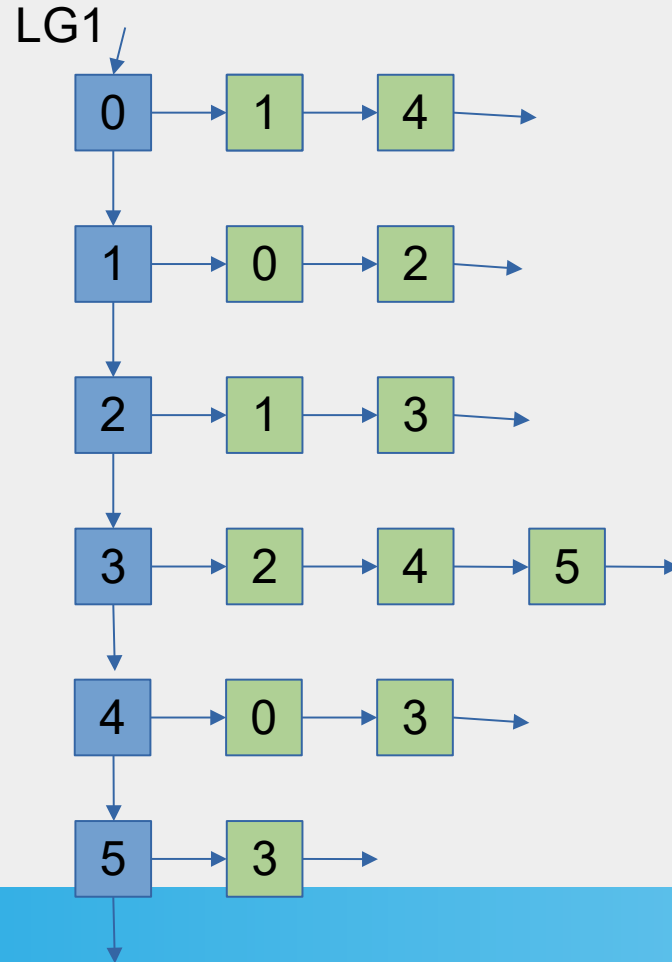
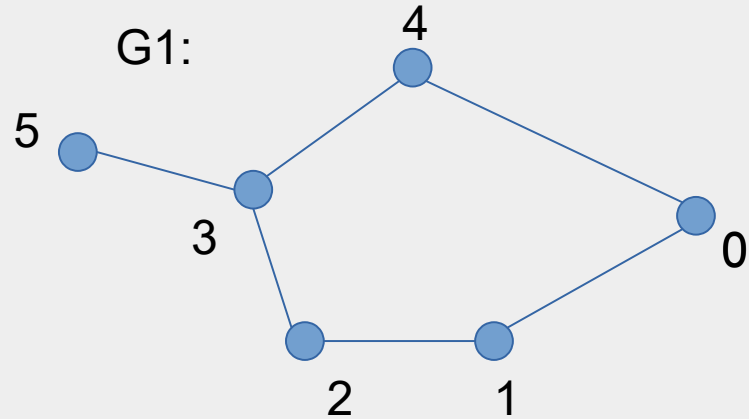


0	1	2	3	4	5
0	0	0	0	0	0

1 2 4 5 3 6

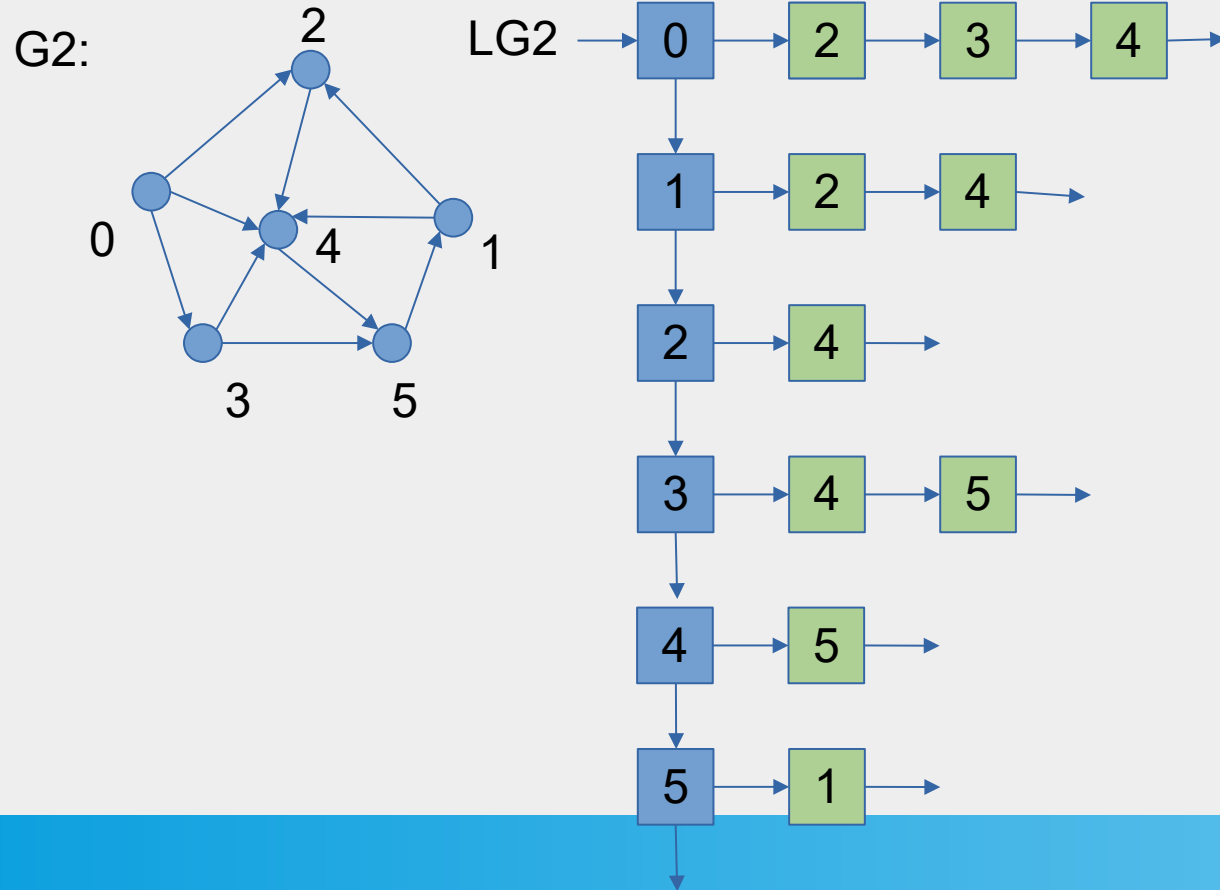
F: 0 1 4
v: 0
F: 1 4 2
v: 1
F: 4 2 3
v: 4
F: 2 3
v: 2
F: 3 5
v: 3
F: 5
v: 5

GSI011 – Estrutura de Dados 2 – Aula 8



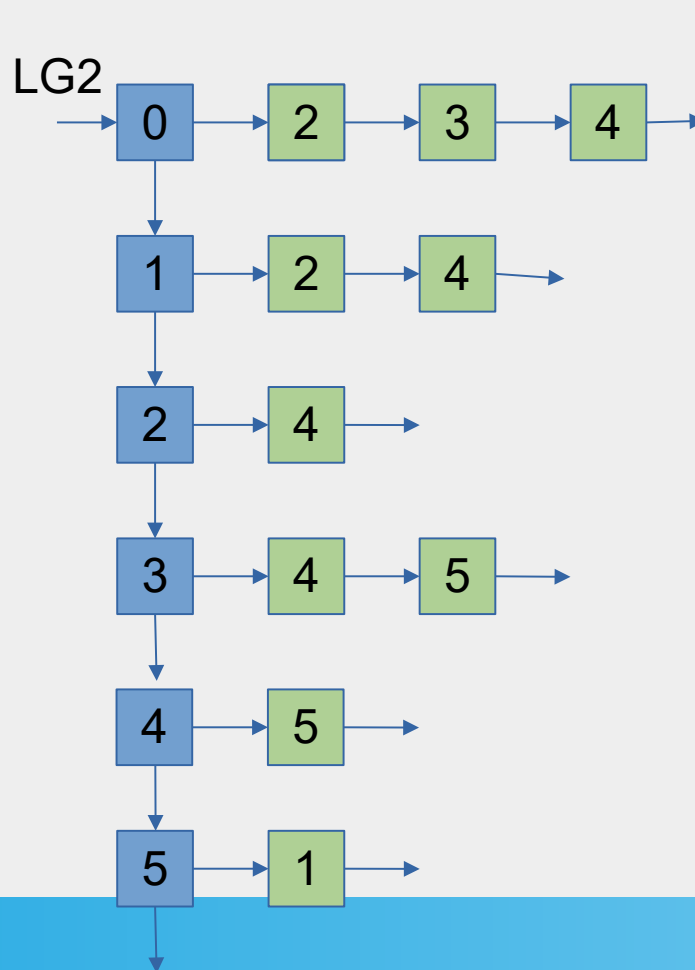
0	1	2	3	4	5
0	0	0	0	0	0
1	2	4	5	3	6
F: 0	1	4			
v: 0					
F: 4	4	2			
v: 1					
F: 4	2	3			
v: 4					
F: 2	3				
v: 2					
F: 3	5				
v: 3					
F: 5					
v: 5					

GSI011 – Estrutura de Dados 2 – Aula 8



GSI011 – Estrutura de Dados 2 – Aula 8

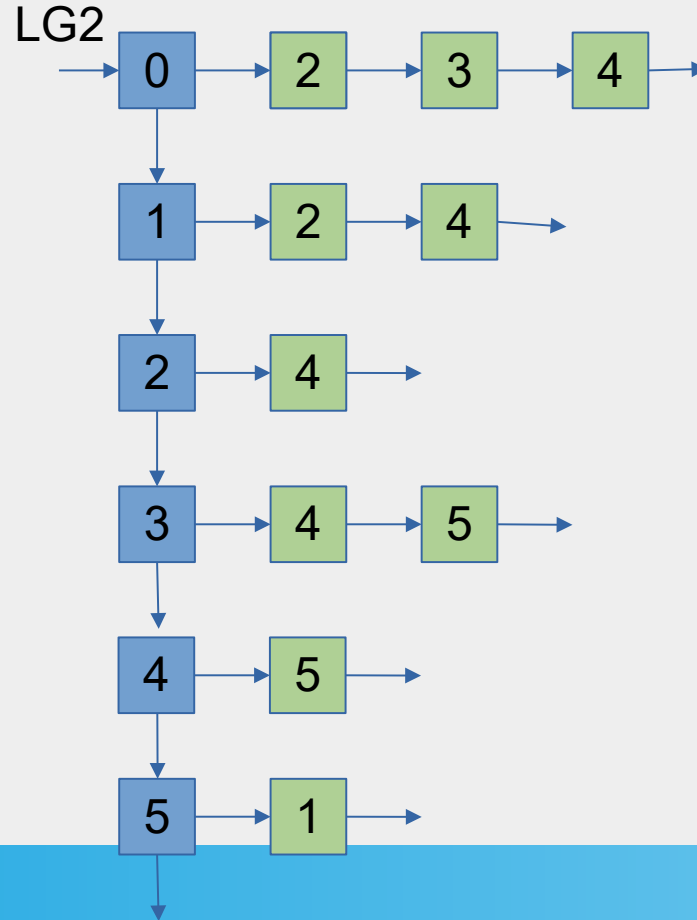
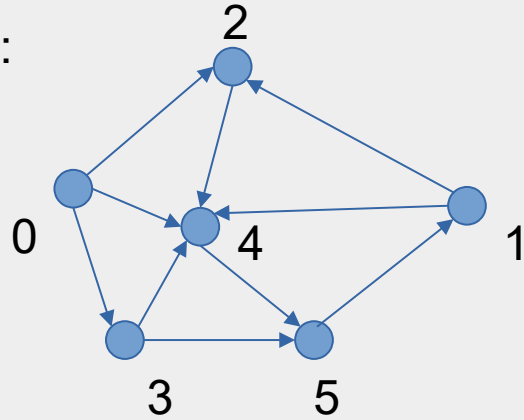
```
BuscaLargura(vi, n, Adj)
INICIO
cont ← 1; // ordem da visitação
para u ← 0 até n – 1 faça visitado[u] ← 0;
visitado[vi] = cont; F ← criaFila(vi);
enquanto (F não vazia) faça
  v ← saiFila(F);
  enquanto houver w em Adj[v] faça
    se (visitado[w] == 0) então
      cont++;
      visitado[w] ← cont;
      entraFila(F, w);
  fimse
fimenquanto
liberaFila(F);
retorna visitado[0...n-1];
FIM
```



0	1	2	3	4	5
0	0	0	0	0	0
1	6	2	3	4	5
F: 0 2 3 4					
v: 0					
F: 2 3 4					
v: 2					
F: 3 4 5					
v: 3					
F: 4 5					
v: 4					
F: 5 1					
v: 5					
F: 4					
v: 1					

GSI011 – Estrutura de Dados 2 – Aula 8

G2:



0	1	2	3	4	5
0	0	0	0	0	0
1	6	2	3	4	5
F: 0	2	3	4		
v: 0					
F: 2	3	4			
v: 2					
F: 3	4	5			
v: 3					
F: 4	5				
v: 4					
F: 5	1				
v: 5					
F: 4					
v: 1					

GSI011 – Estrutura de Dados 2 – Aula 8

- Busca em Profundidade:
 - Inicia-se em um vértice v_i e aprofunda-se nos vértices vizinhos dele até que:
 - O alvo seja encontrado
 - Não existam mais vértices a serem visitados
 - Para realizar essa tarefa é utilizado o mecanismo de ***backtracking*** (recursão).

GSI011 – Estrutura de Dados 2 – Aula 8

- Busca em Profundidade:**

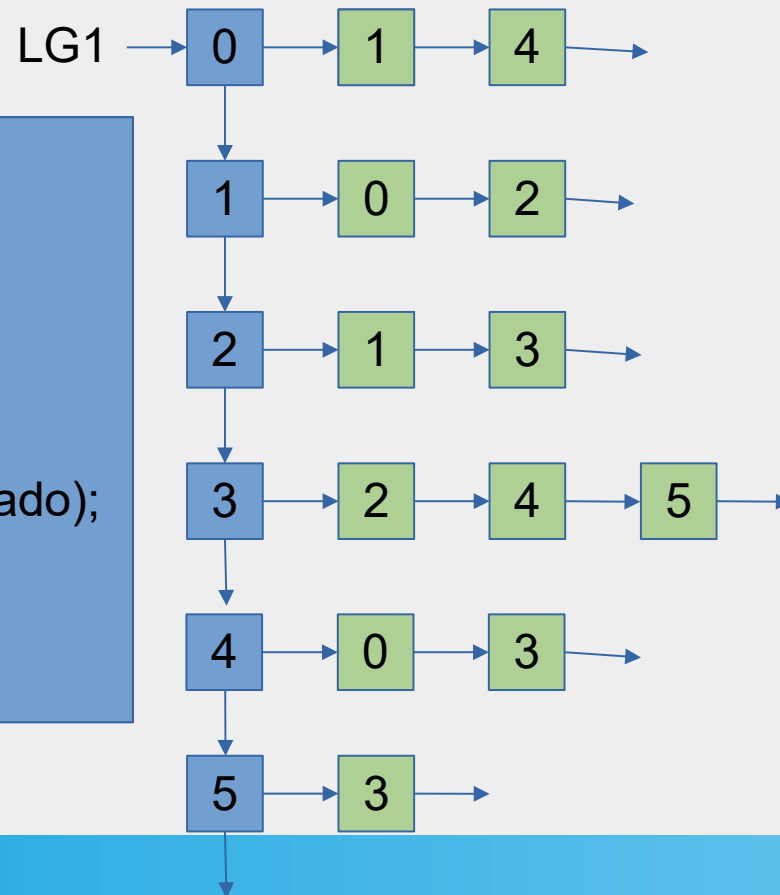
```
BuscaProfundidade(vi, n, Adj)
INICIO
cont ← 0; //variável global
para u ← 0 até n – 1 faça visitado[u] ← 0;
Profundidade (vi, Adj, visitado);
retorna visitado[0...n-1];
FIM
```

```
Profundidade(v, Adj, visitado)
INICIO
cont++; //variável global
visitado[v] ← cont;
enquanto houver w em Adj[v] faça
    se (visitado[w] == 0) então
        Profundidade(w, Adj, visitado);
    fimse
fimenquanto
FIM
```

Complexidade: $O(n + m)$

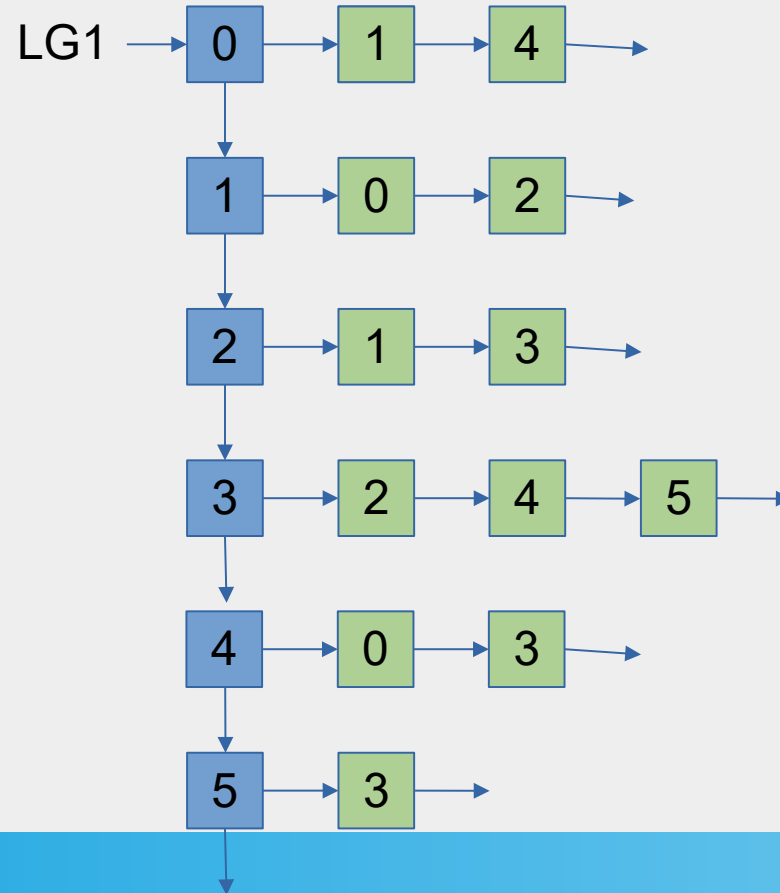
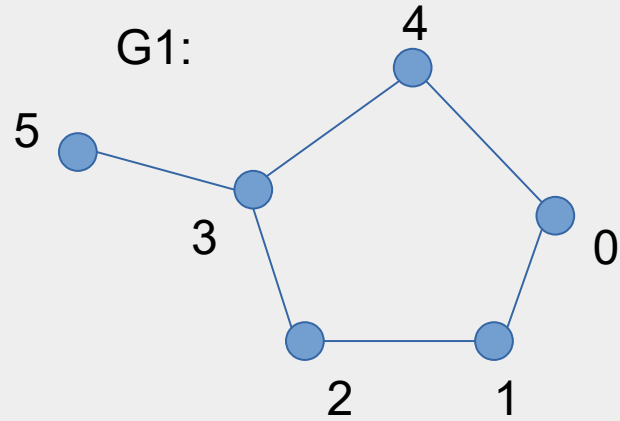
GSI011 – Estrutura de Dados 2 – Aula 8

```
Profundidade(v, Adj,visitado)
INICIO
cont++;//variável global
visitado[v] ← cont;
enquanto houver w em Adj[v] faça
    se (visitado[w] == 0) então
        Profundidade(w, Adj, visitado);
    fimse
fimenquanto
FIM
```



0	1	2	3	4	5
0	0	0	0	0	0
1	2	3	4	5	6

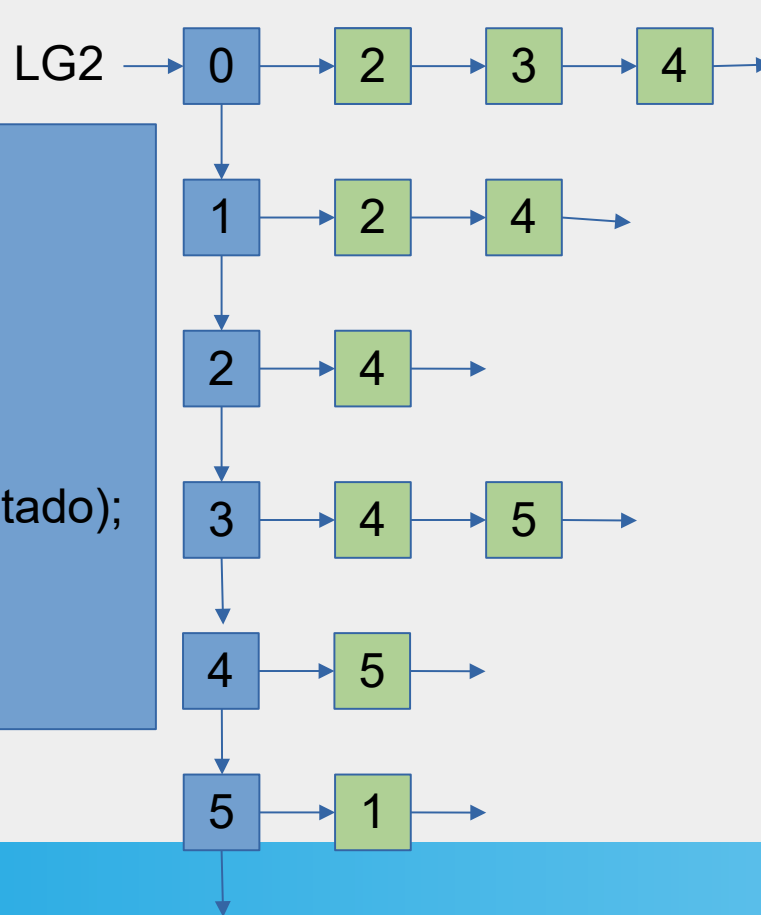
GSI011 – Estrutura de Dados 2 – Aula 8



0	1	2	3	4	5
0	0	0	0	0	0
1	2	3	4	5	6

GSI011 – Estrutura de Dados 2 – Aula 8

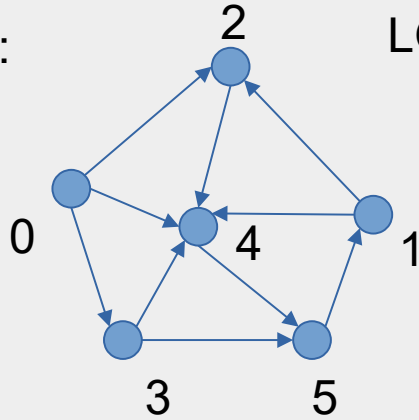
```
Profundidade(v, Adj,visitado)
INICIO
cont++;//variável global
visitado[v] ← cont;
enquanto houver w em Adj[v] faça
    se (visitado[w] == 0) então
        Profundidade(w, Adj, visitado);
    fimse
fimenquanto
FIM
```



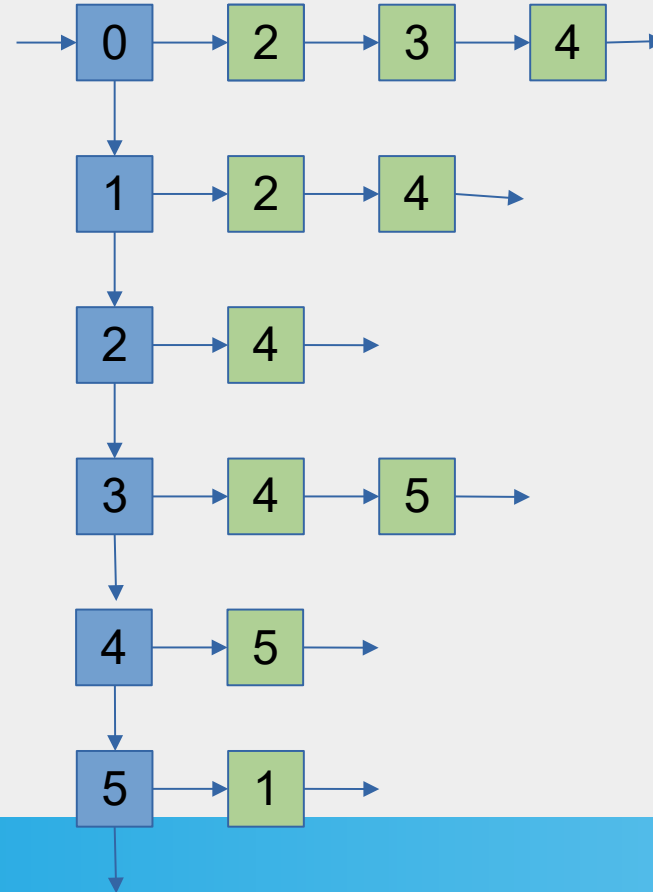
0	1	2	3	4	5
0	0	0	0	0	0
1	5	2	6	3	4

GSI011 – Estrutura de Dados 2 – Aula 8

G2:



LG2



0	1	2	3	4	5
0	0	0	0	0	0
1	5	2	6	3	4