Depth First Search
(C++)

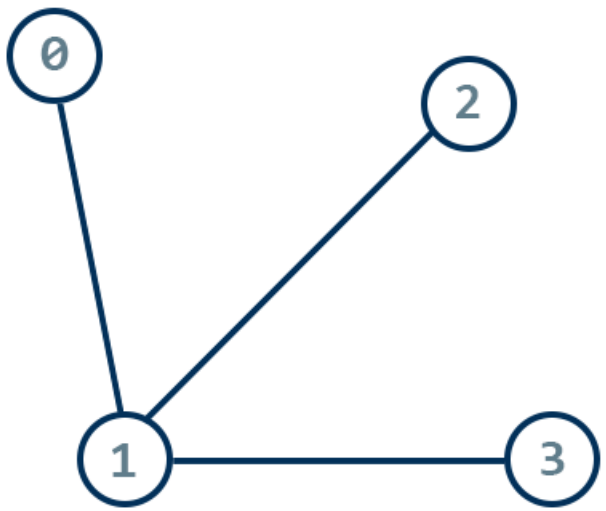
#csspree Online

Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.

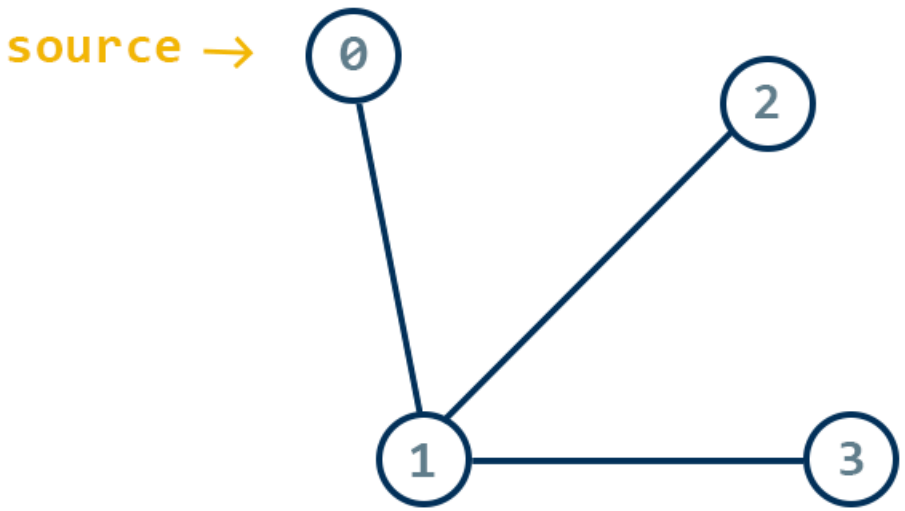
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



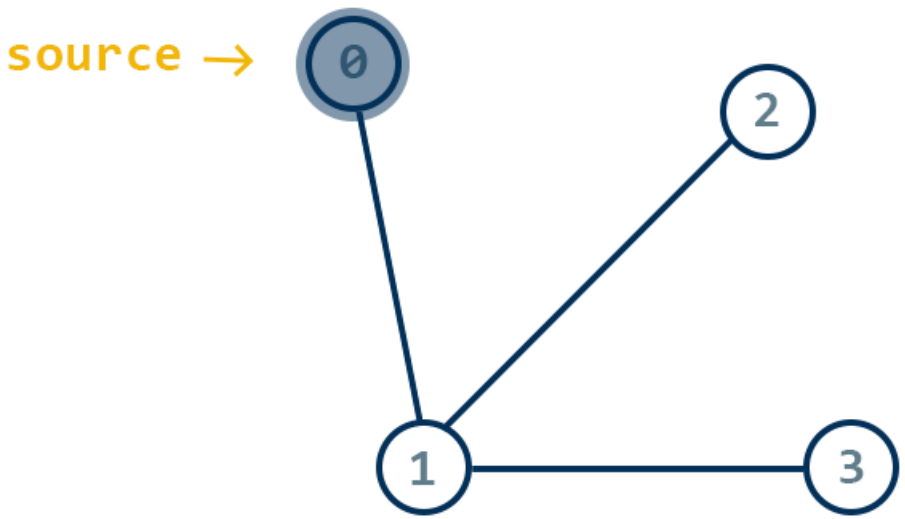
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



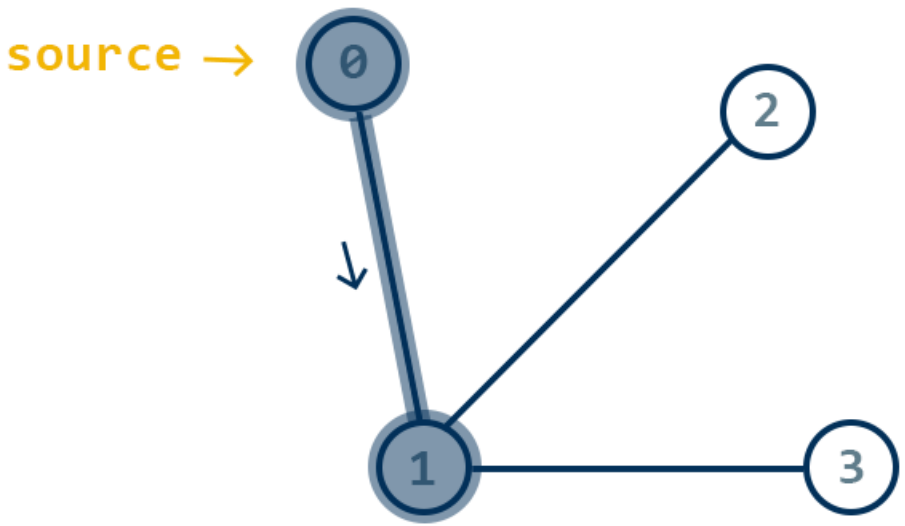
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



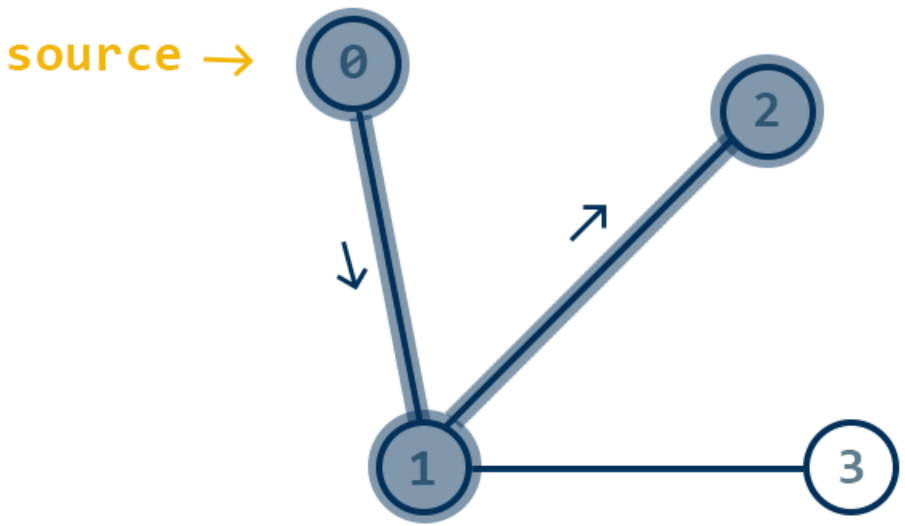
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



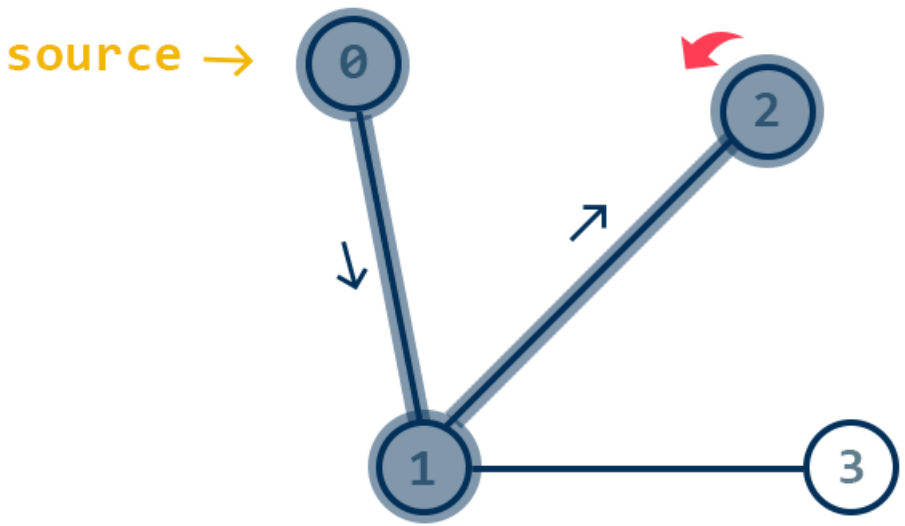
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



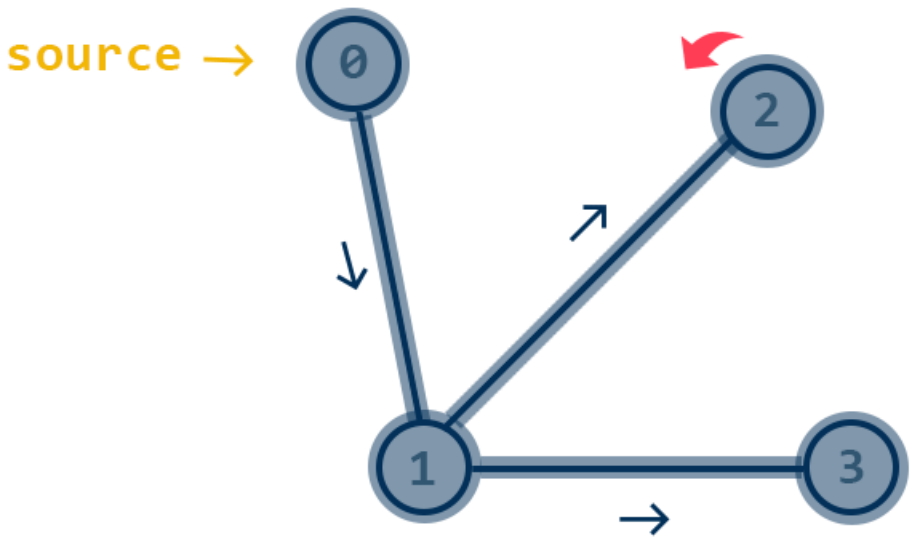
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



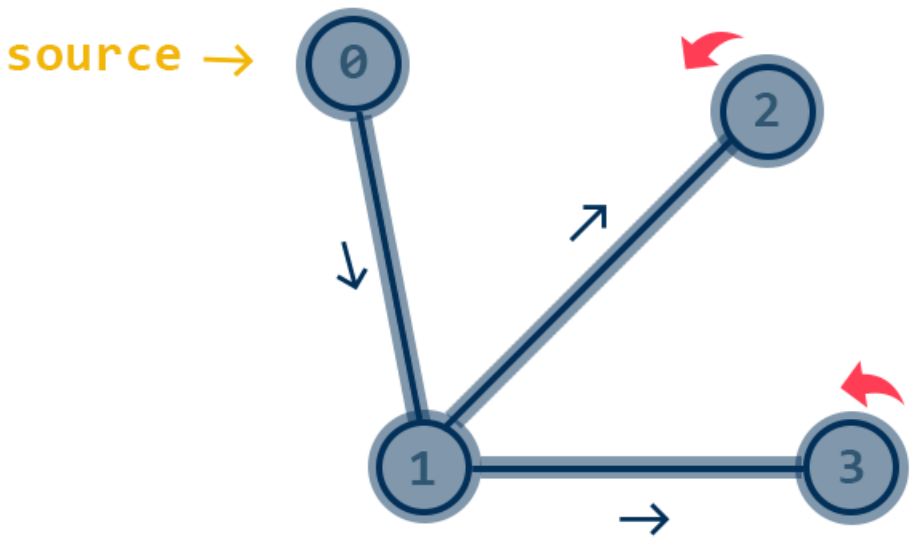
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



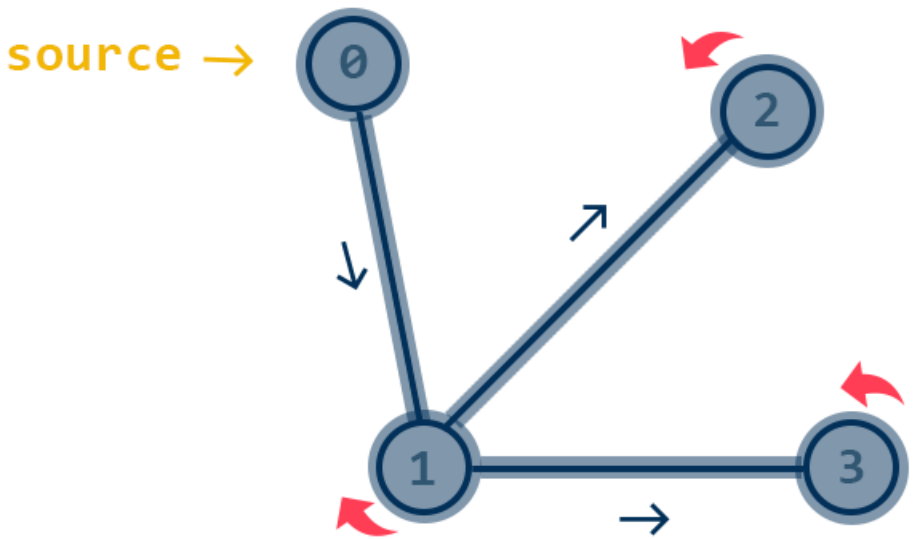
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



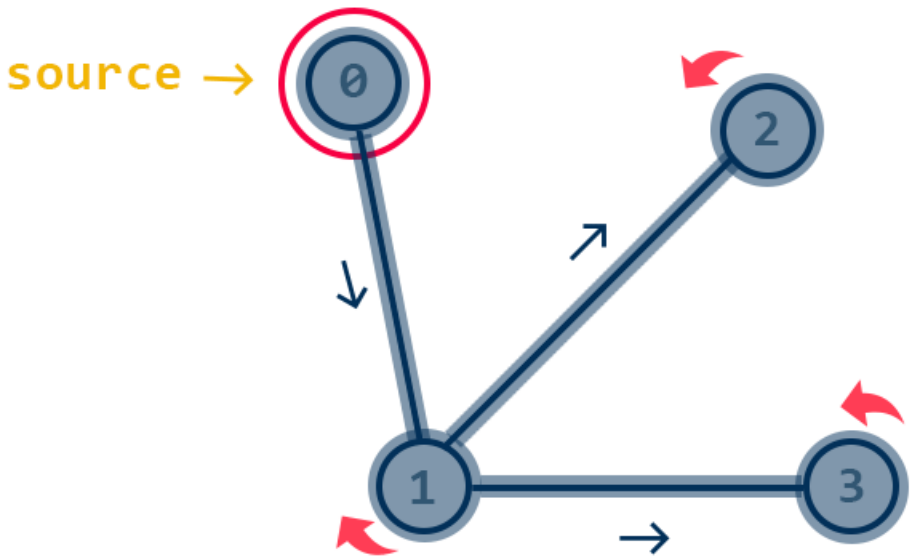
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



Depth First Search (DFS)

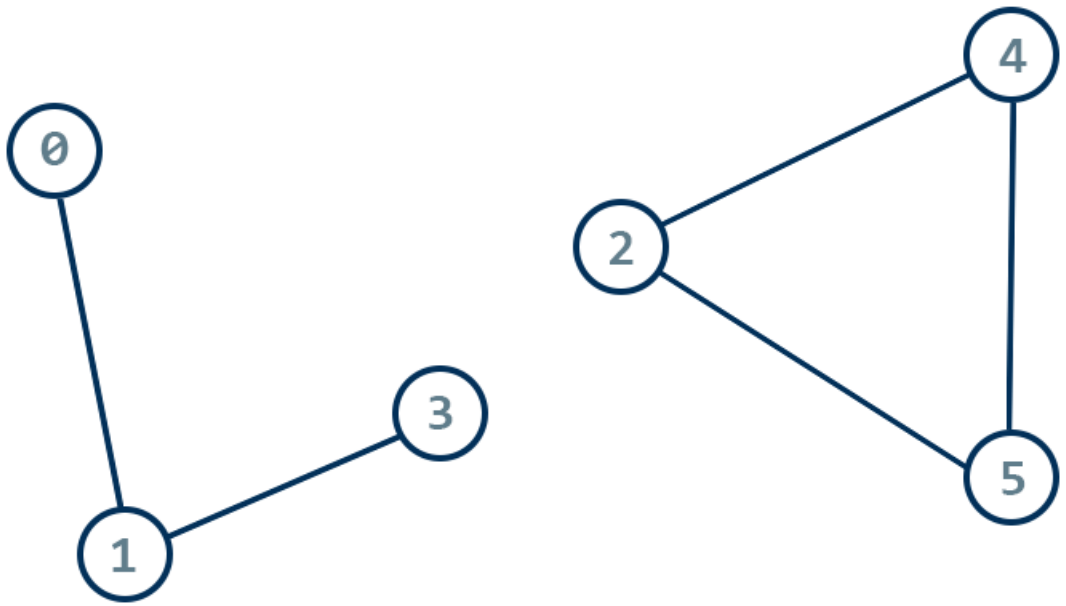
It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



when the dfs is back to the starting source, the "traversal" for that call is done!

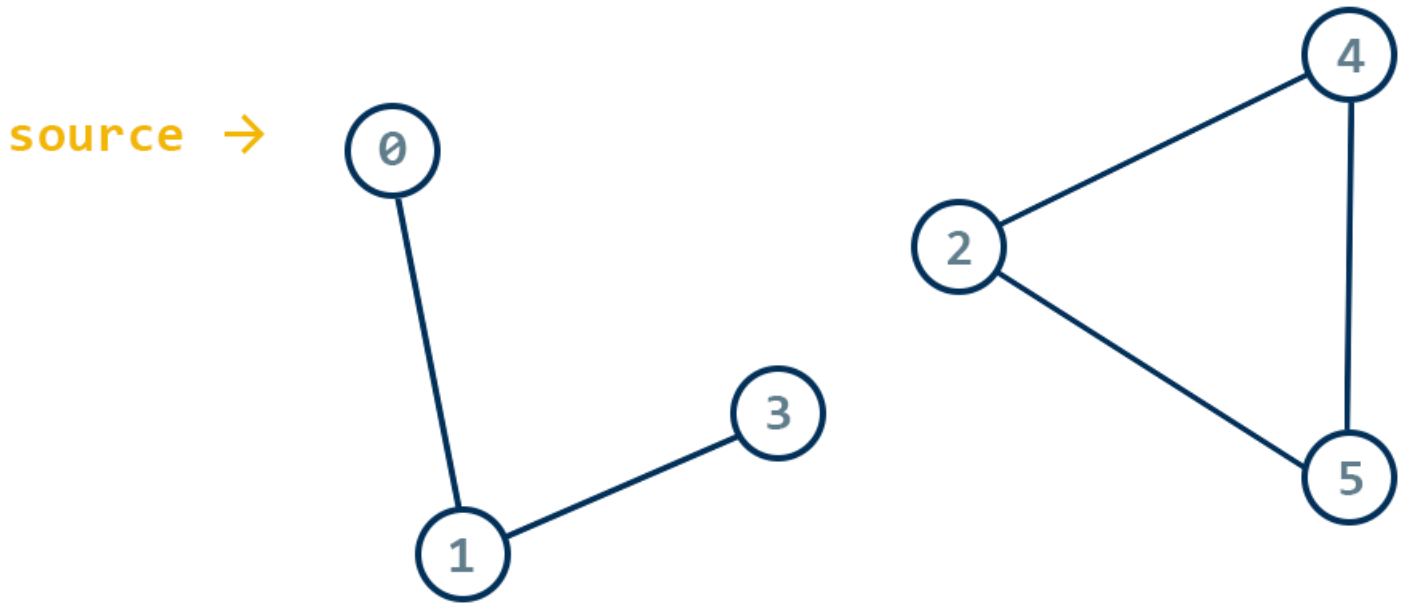
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



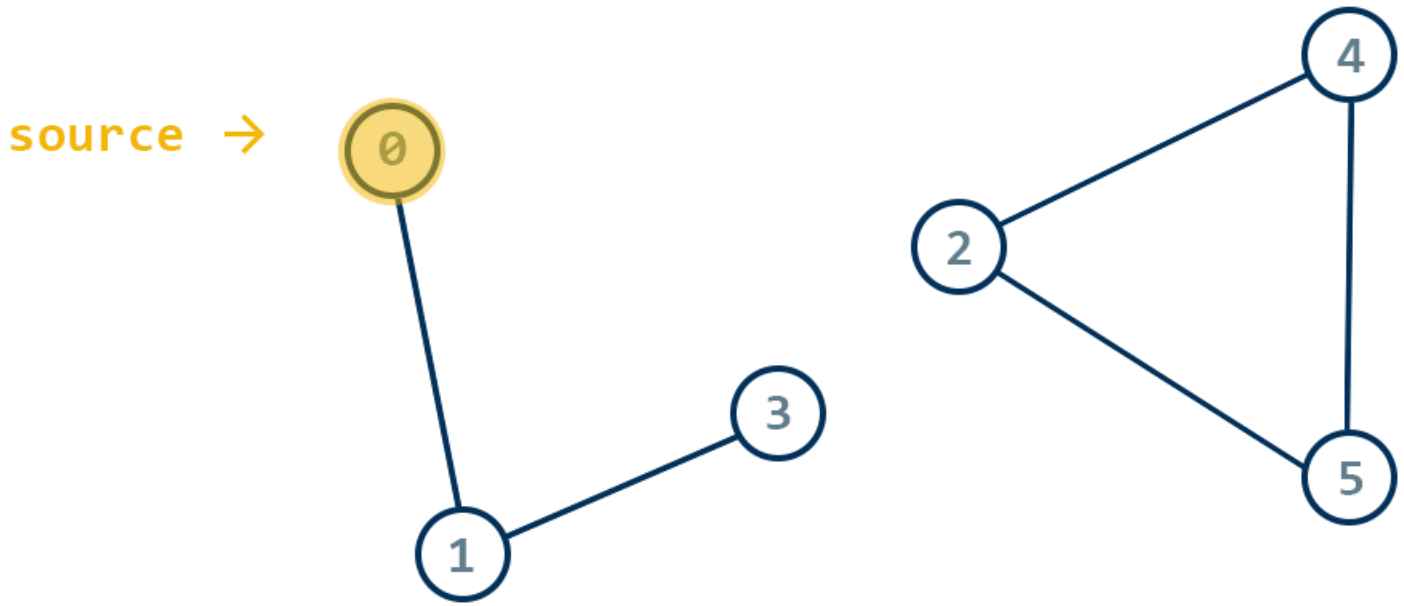
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



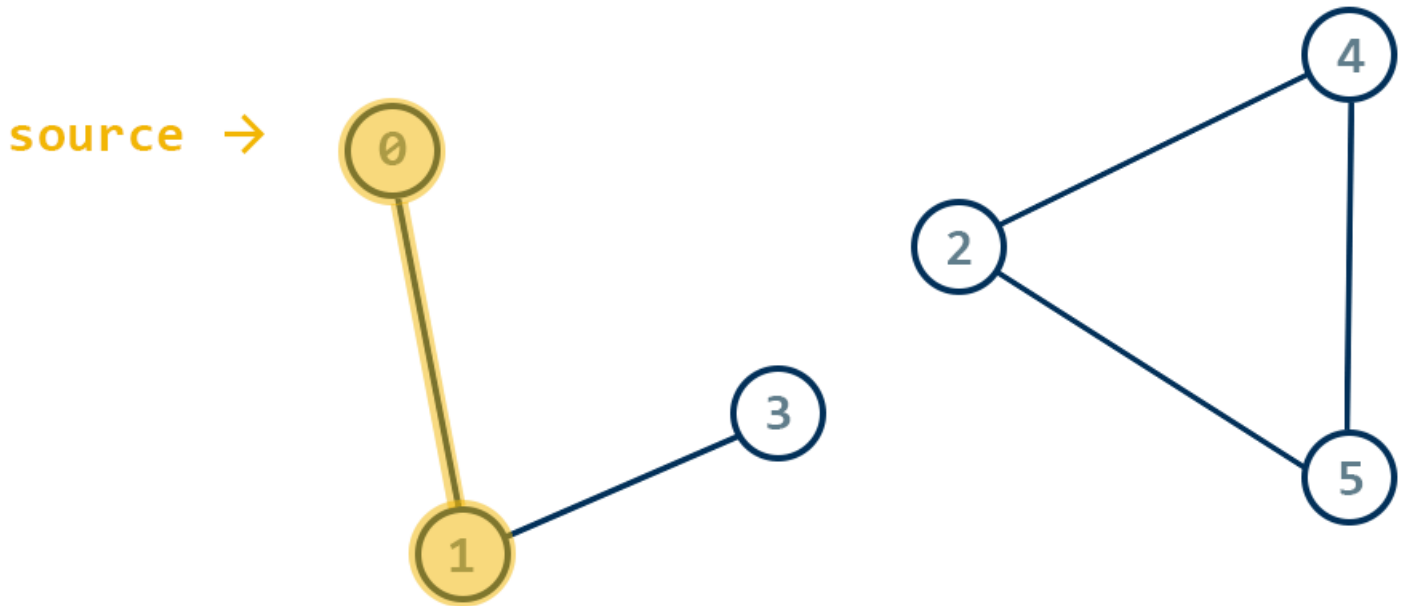
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



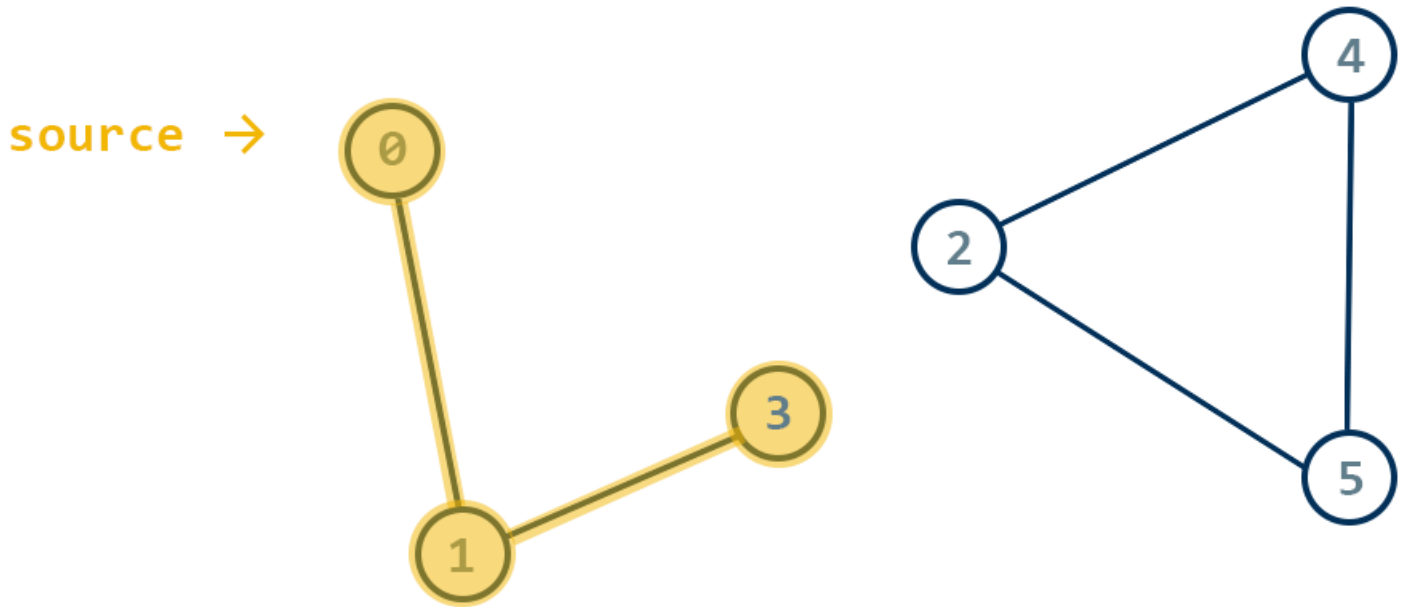
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



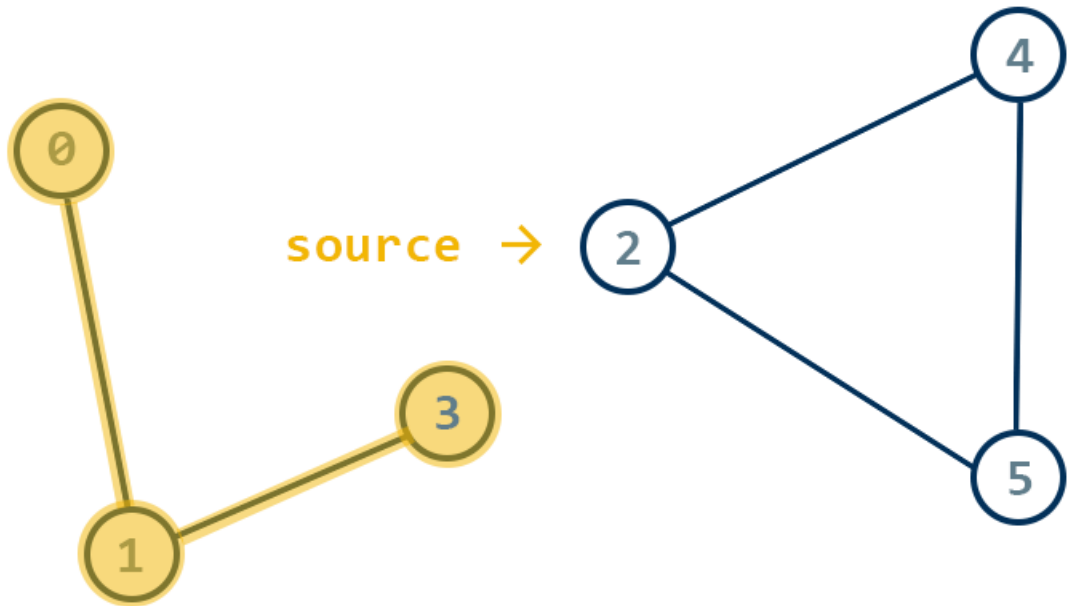
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



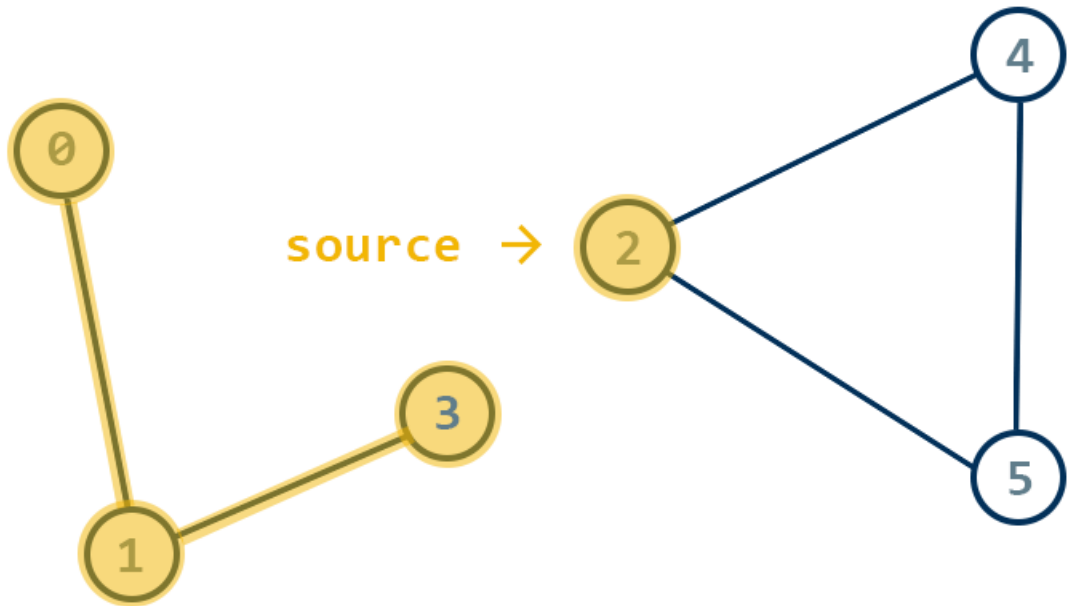
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



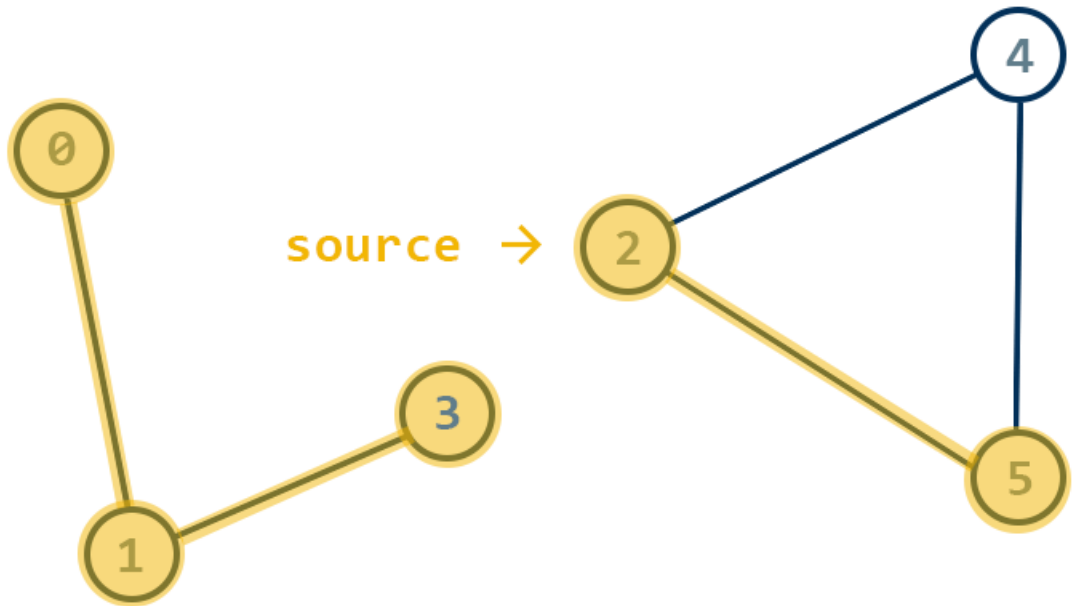
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



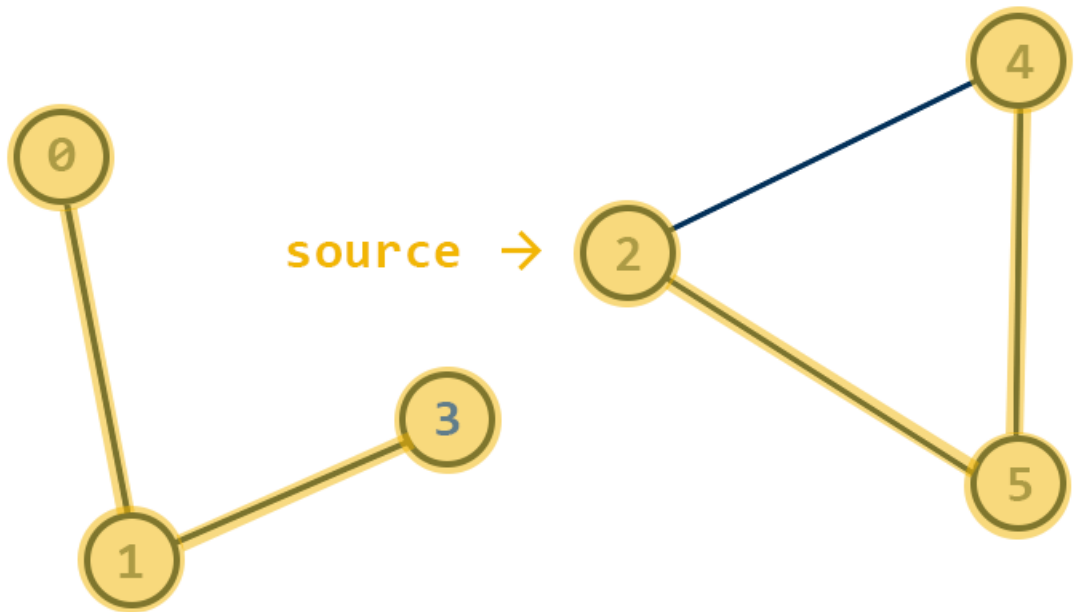
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



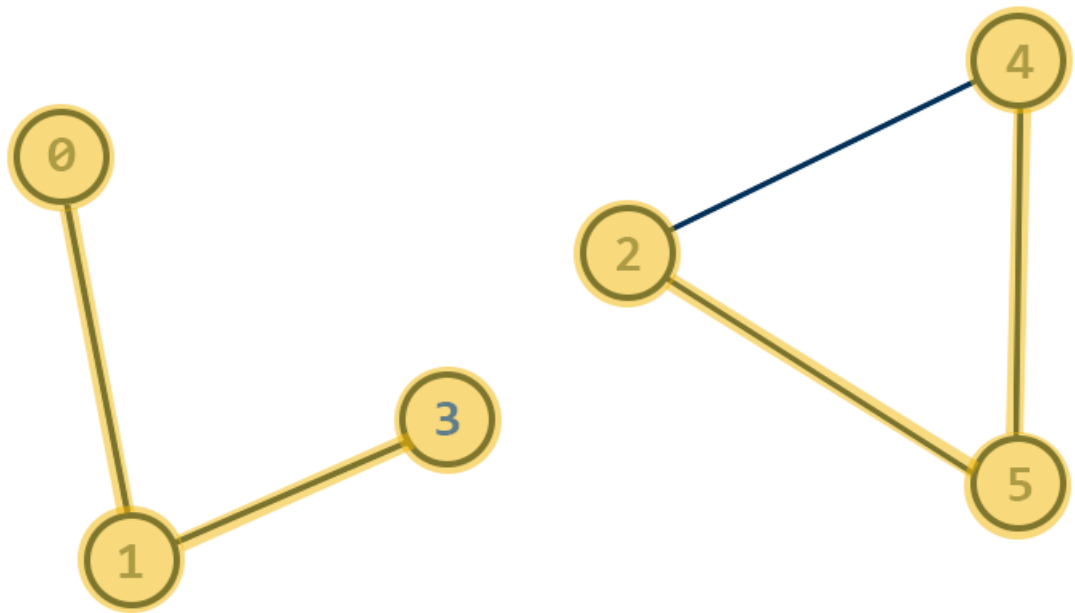
Depth First Search (DFS)

It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



Depth First Search (DFS)

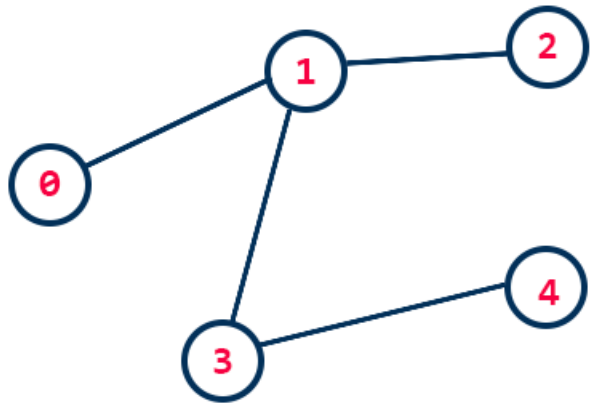
It is a graph traversal algorithm using backtracking. It goes as deep as possible before it backtracks.



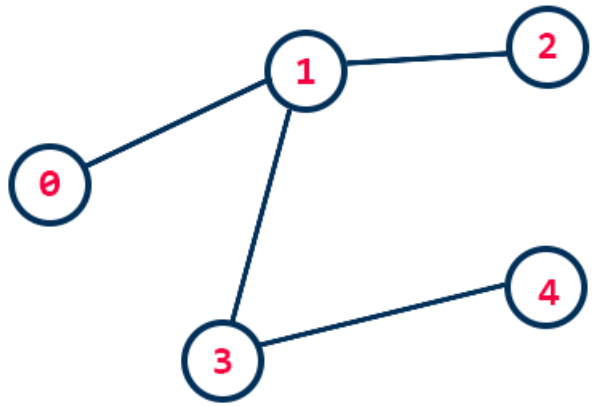

```
Total dfs call = 2
```

DFS Simulation

DFS Simulation

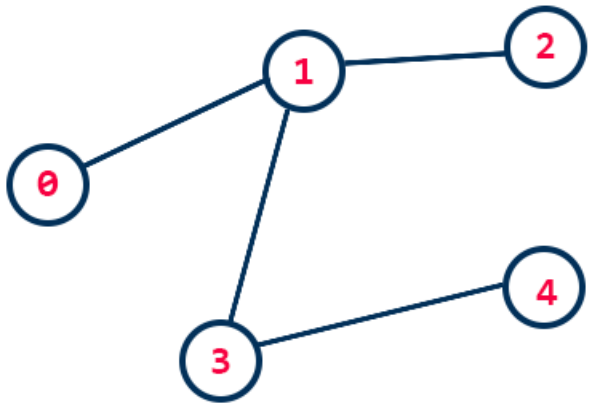


DFS Simulation

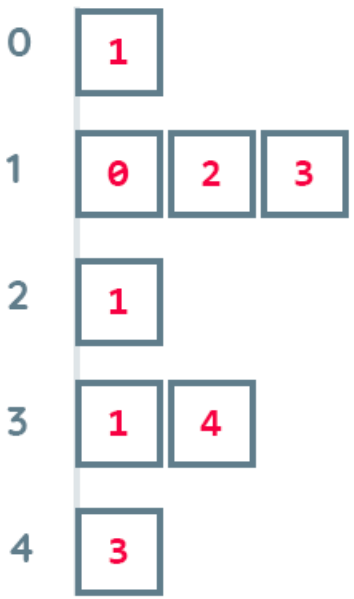


0	1
1	0, 2, 3
2	1
3	1, 4
4	3

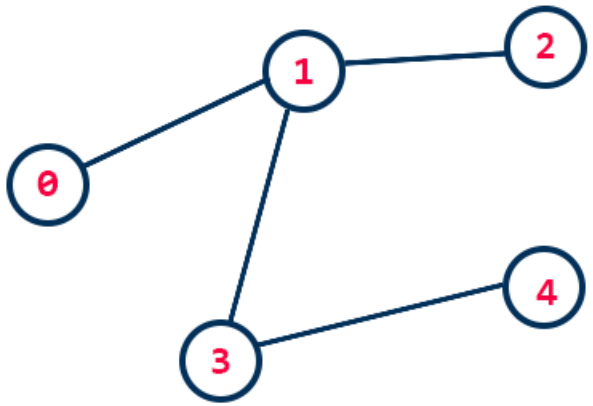
DFS Simulation




```
bool visited[5];
```

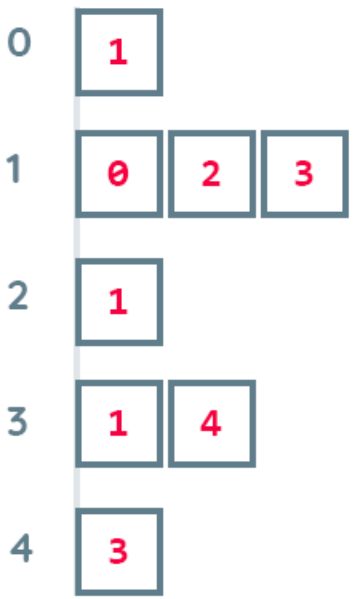


DFS Simulation

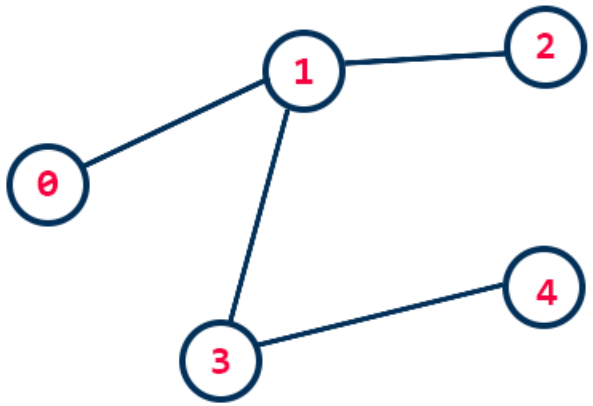



```
bool visited[5];
```

0	0	0	0	0
0	1	2	3	4

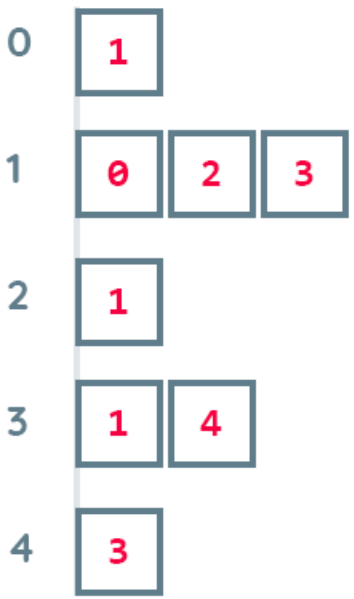


DFS Simulation



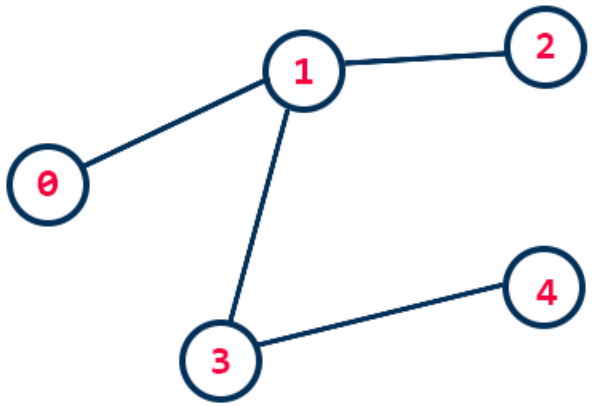

```
bool visited[5];
```

0	0	0	0	0
0	1	2	3	4



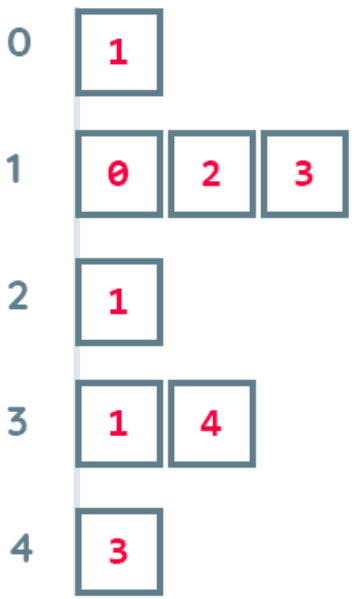

```
nodes: 0  1  2  3  4
```

DFS Simulation




```
bool visited[5];
```

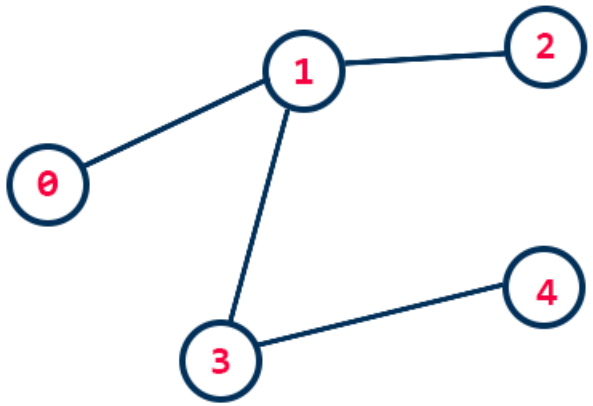
0	0	0	0	0
0	1	2	3	4

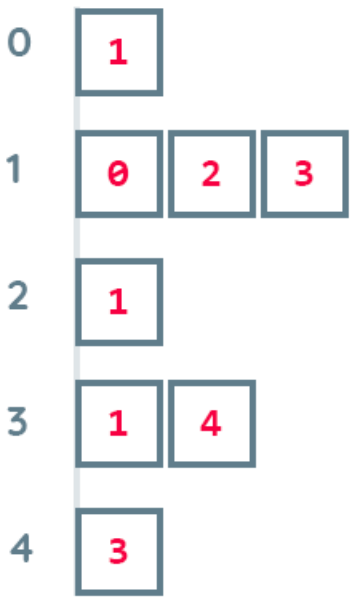



```
nodes: 0  1  2  3  4

source:
```

DFS Simulation






```
bool visited[5];
```

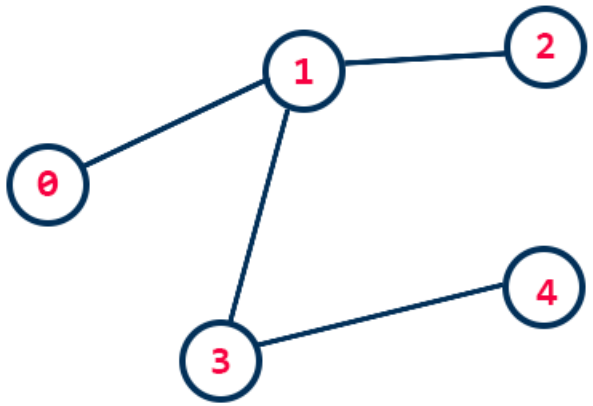
0	0	0	0	0
0	1	2	3	4

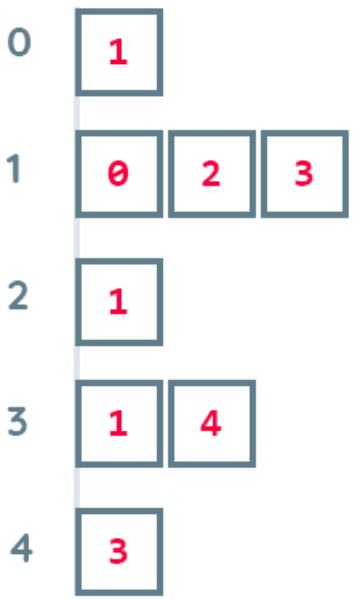
nodes: 0 1 2 3 4

source:

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

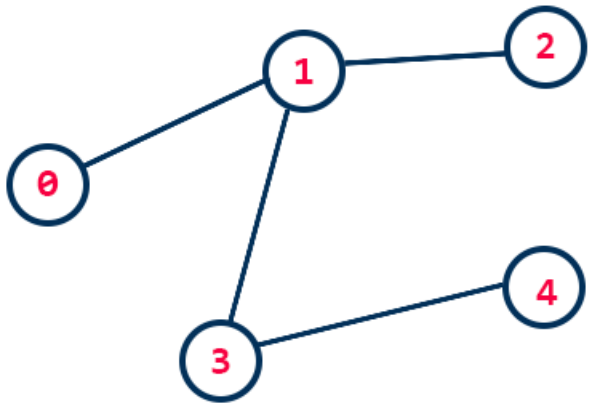
0	0	0	0	0
0	1	2	3	4

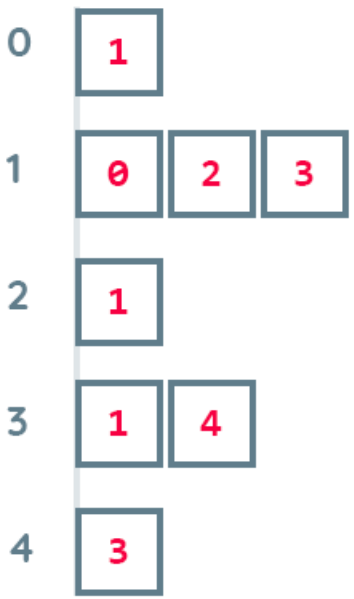
nodes: 0 1 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	0	0	0	0
0	1	2	3	4

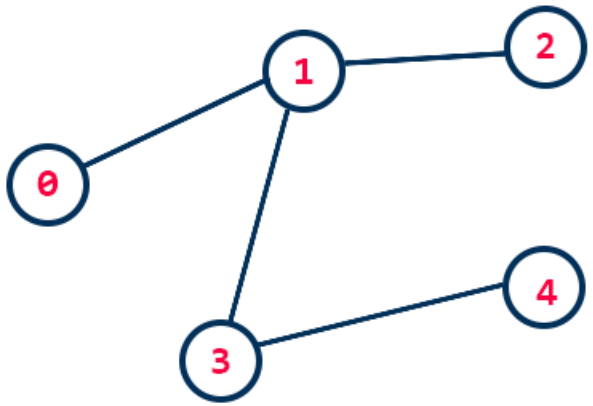
```
// 1 means that index(node)
// is visited
```

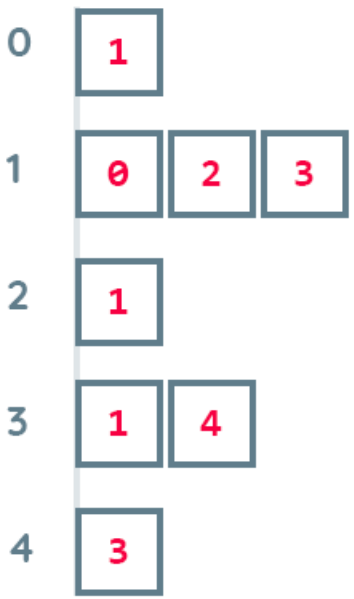
nodes: 0 1 2 3 4

source: 0

```
dfs(source):
  visited[source] = 1
  for next in graph[source]:
    if not visited[next]:
      dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	0	0	0	0
0	1	2	3	4

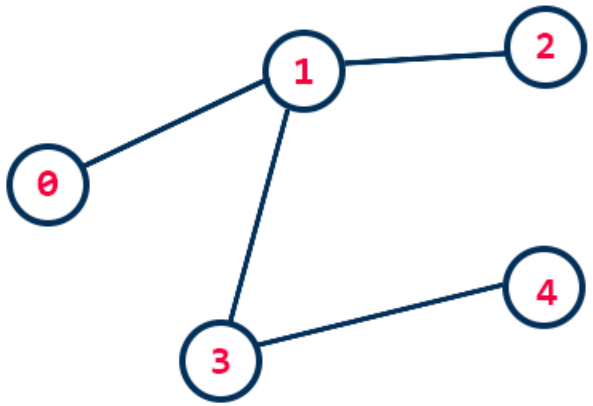
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

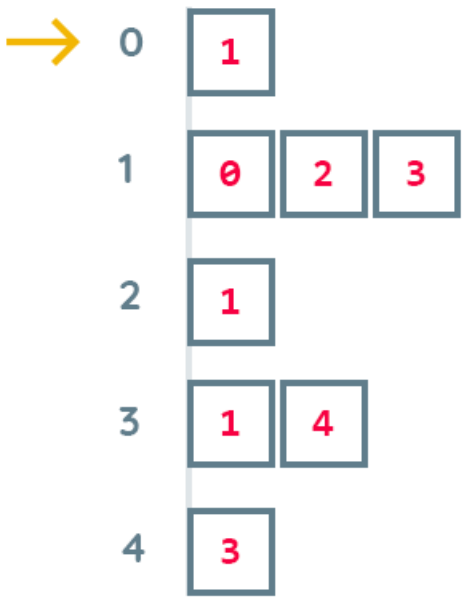
source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation



graph[0]




```
bool visited[5];
```

1	0	0	0	0
0	1	2	3	4

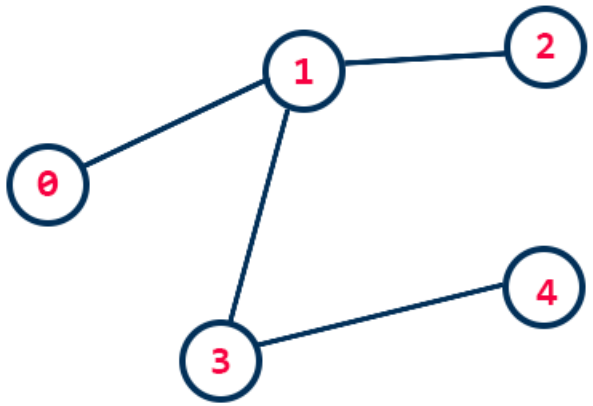
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

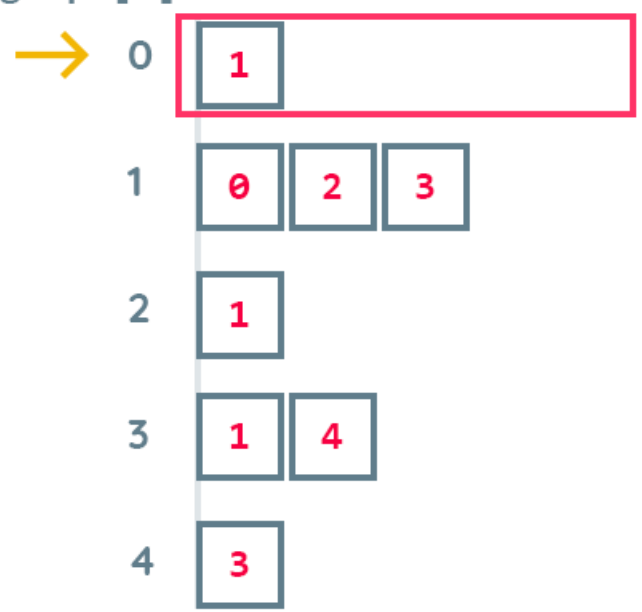
source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation



graph[0]




```
bool visited[5];
```

1	0	0	0	0
0	1	2	3	4

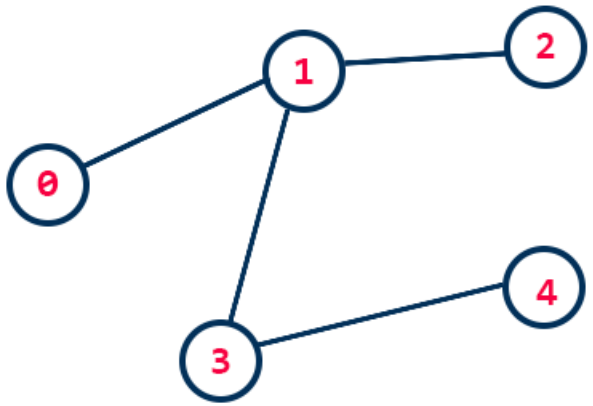
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

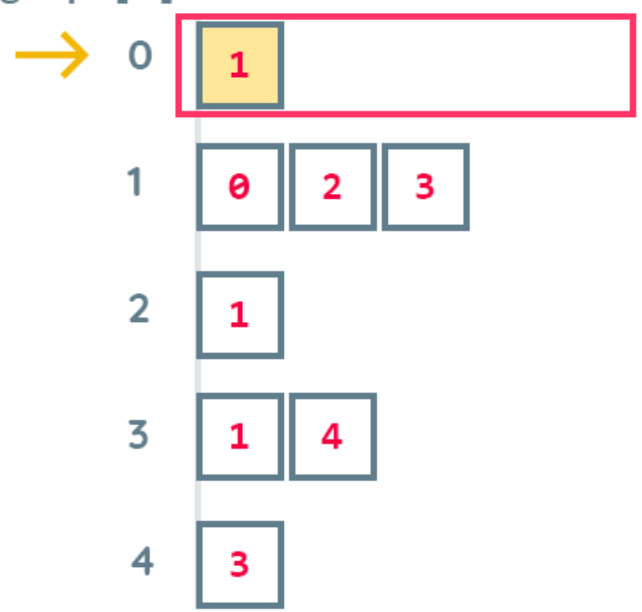
source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation



graph[0]




```
bool visited[5];
```

1	0	0	0	0
0	1	2	3	4

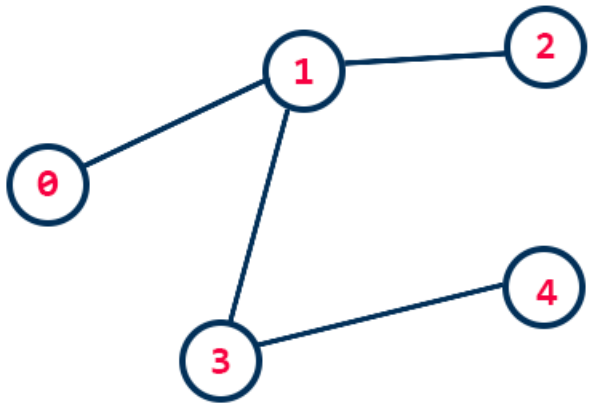
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation



graph[0]

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

```
bool visited[5];
```

1	0	0	0	0
0	1	2	3	4

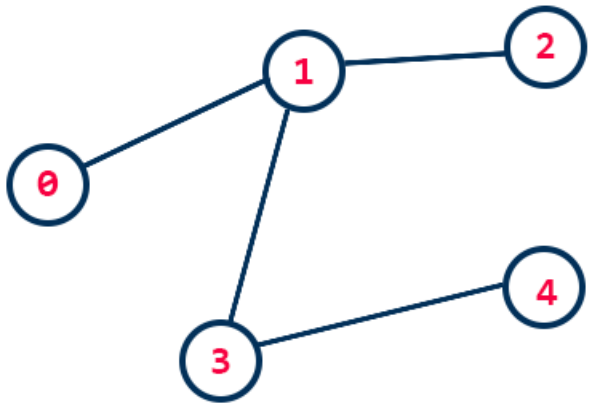
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

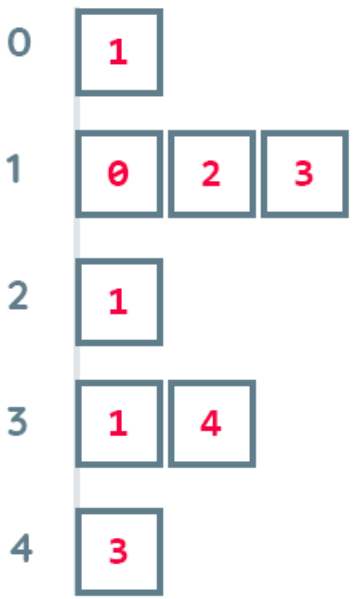
DFS Simulation




```
bool visited[5];
```

1	0	0	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

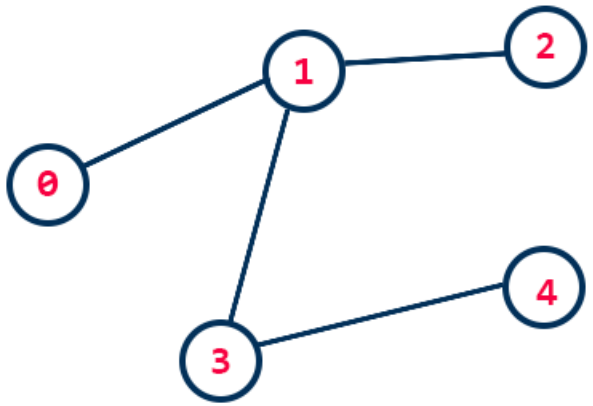


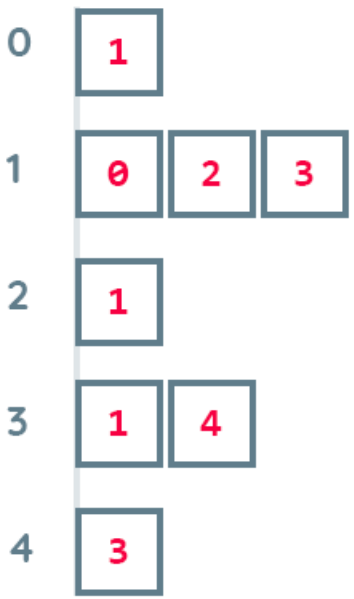
nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	0	0	0
0	1	2	3	4

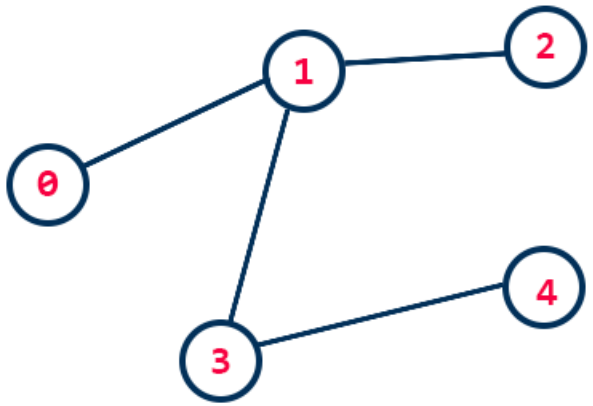
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

source: 1

```
dfs(source):
  visited[source] = 1
  for next in graph[source]:
    if not visited[next]:
      dfs(next)
```

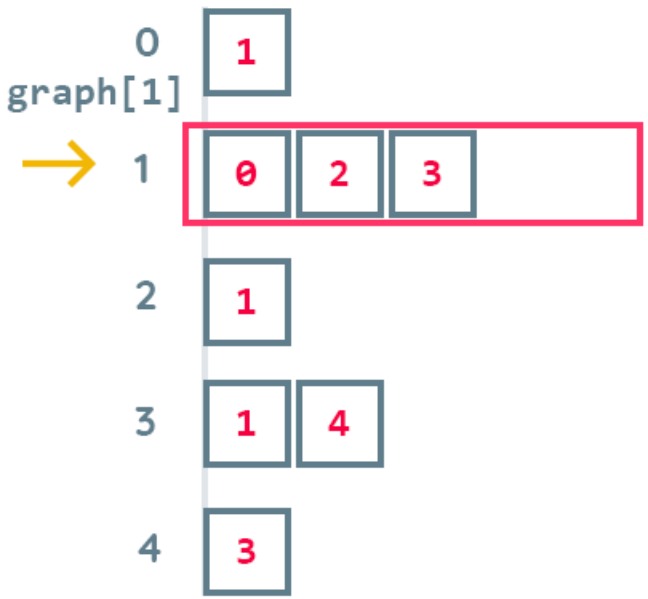
DFS Simulation




```
bool visited[5];
```

1	1	0	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

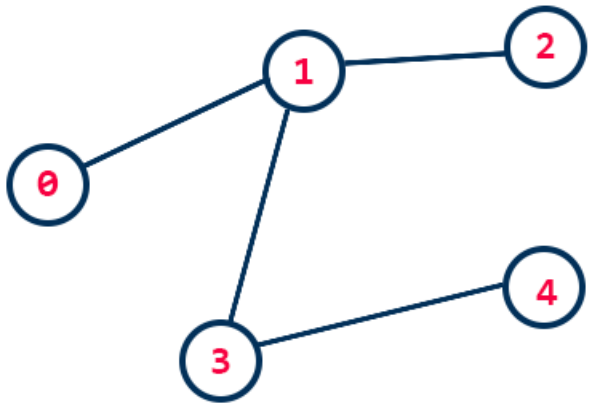


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

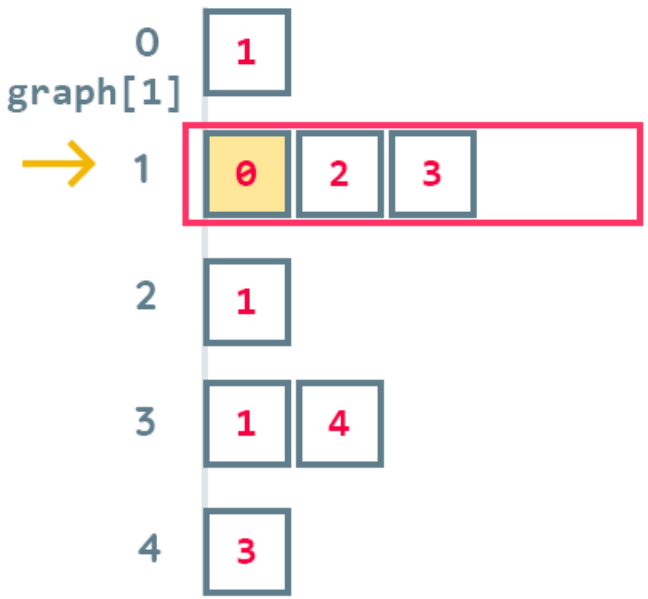
DFS Simulation




```
bool visited[5];
```

1	1	0	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

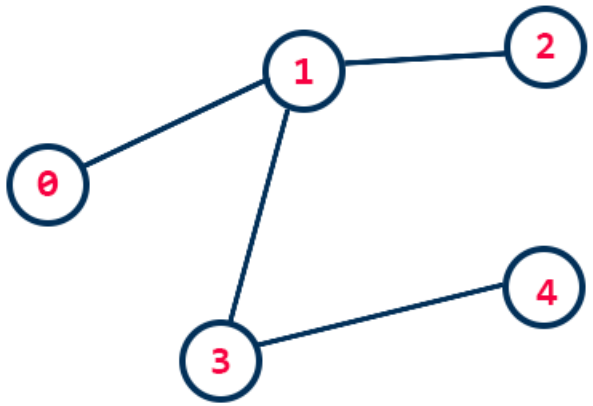


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

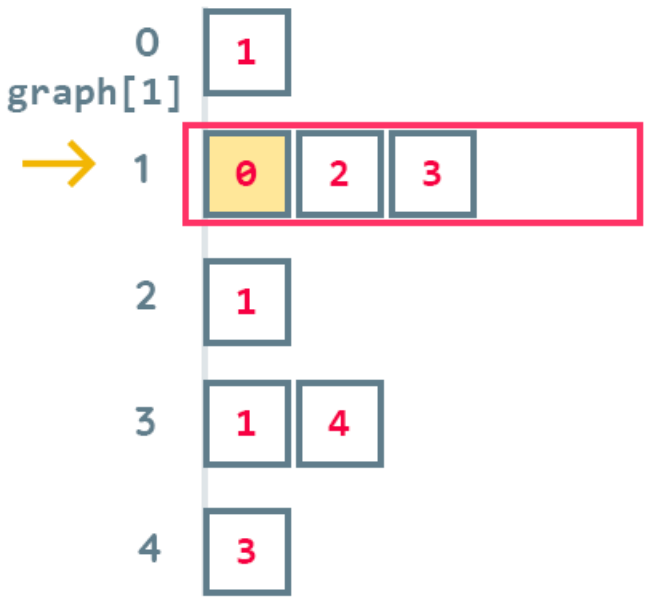
DFS Simulation




```
bool visited[5];
```

1	1	0	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

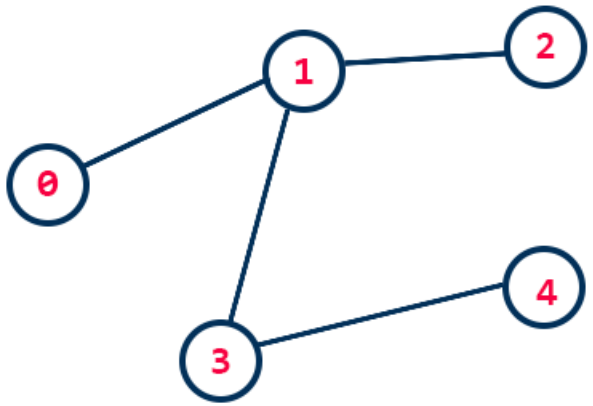


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

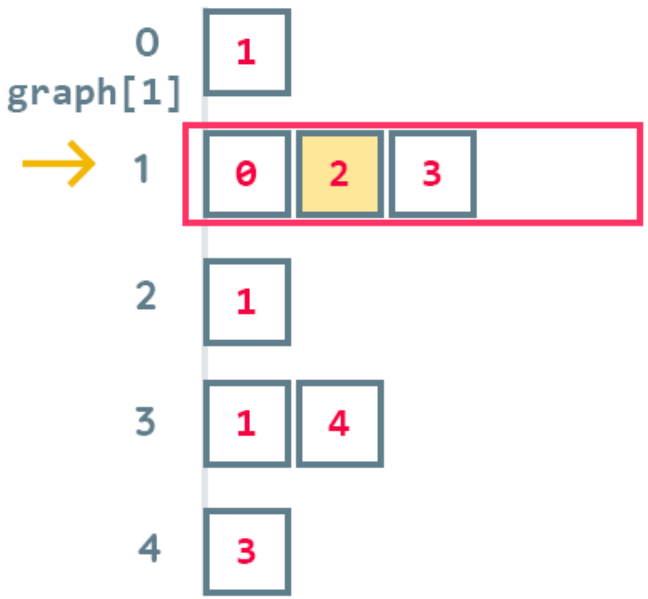
DFS Simulation




```
bool visited[5];
```

1	1	0	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

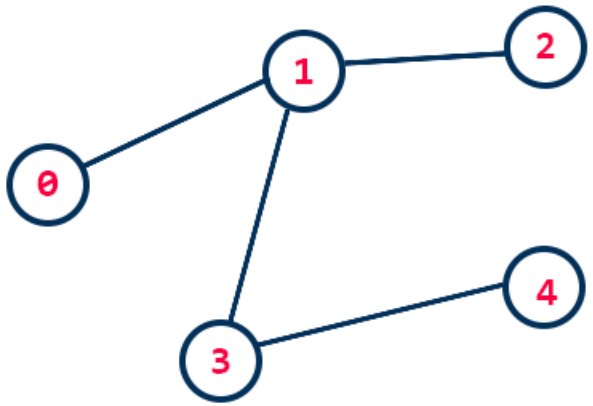


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

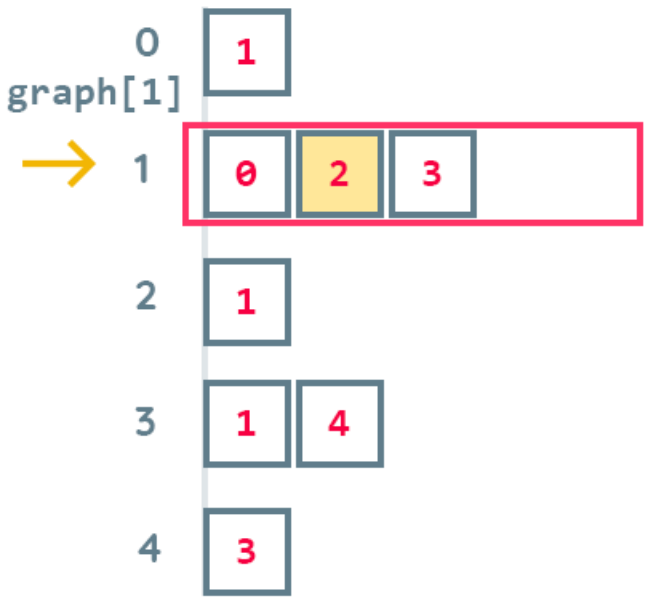
DFS Simulation




```
bool visited[5];
```

1	1	0	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

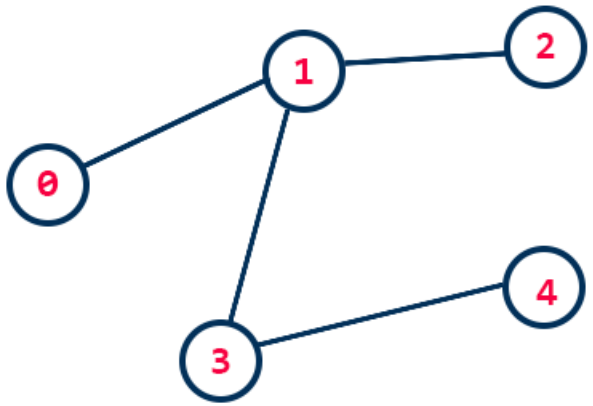


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

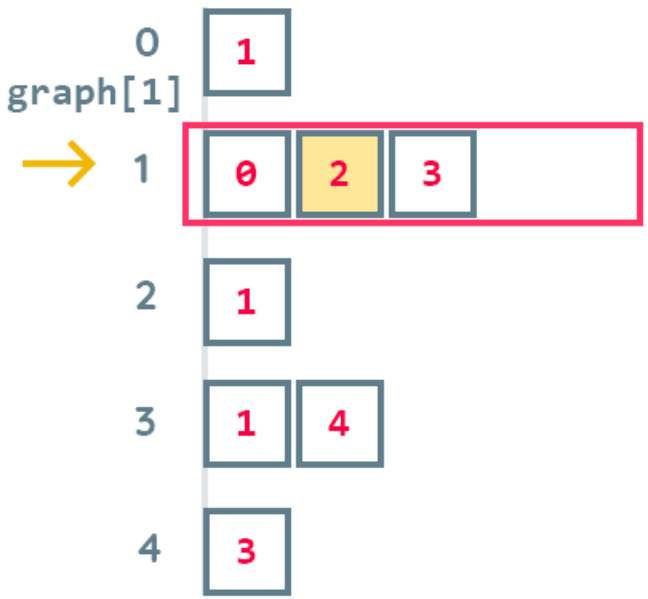
DFS Simulation




```
bool visited[5];
```

1	1	0	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

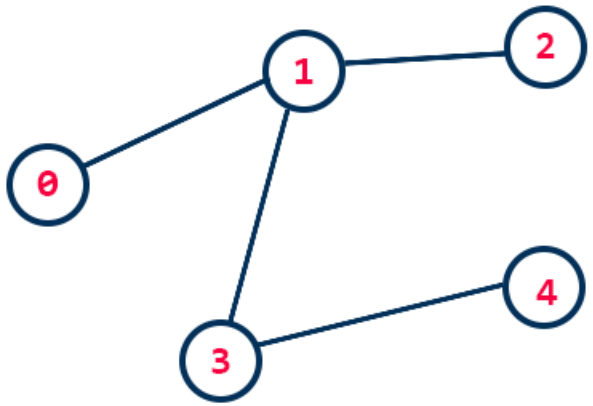


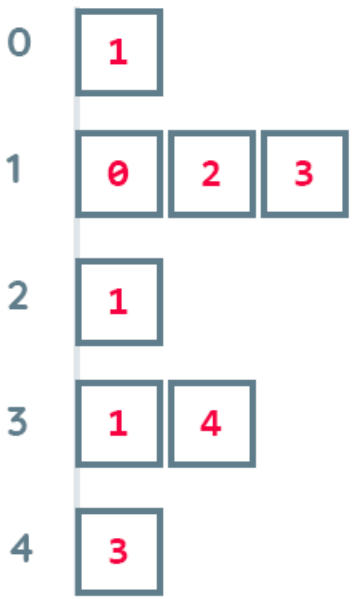
nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	0	0	0
0	1	2	3	4

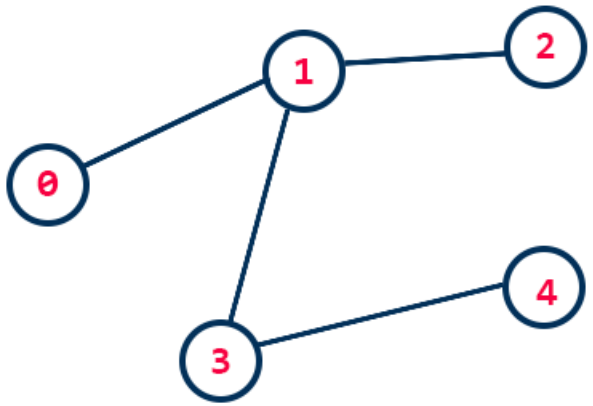
```
// 1 means that index(node)
// is visited
```

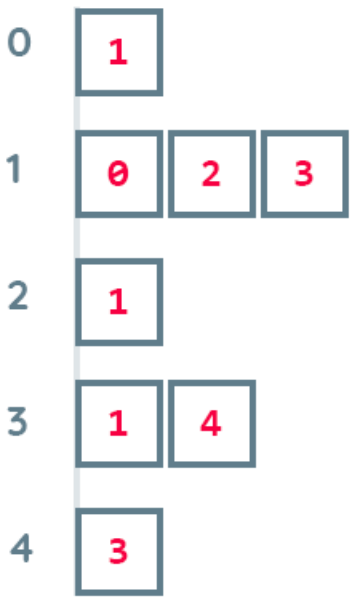
nodes: 0 1 2 3 4

source: 2

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

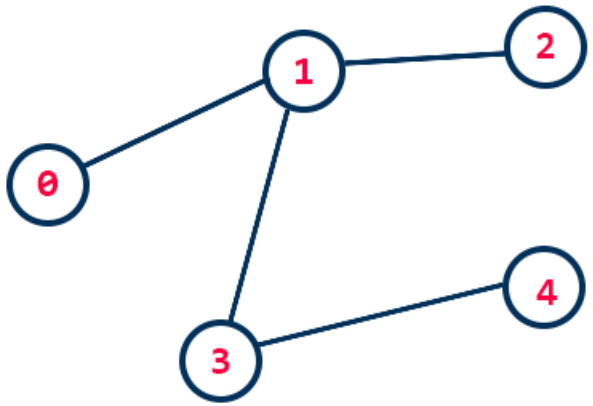
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

source: 2

```
dfs(source):
  visited[source] = 1
  for next in graph[source]:
    if not visited[next]:
      dfs(next)
```

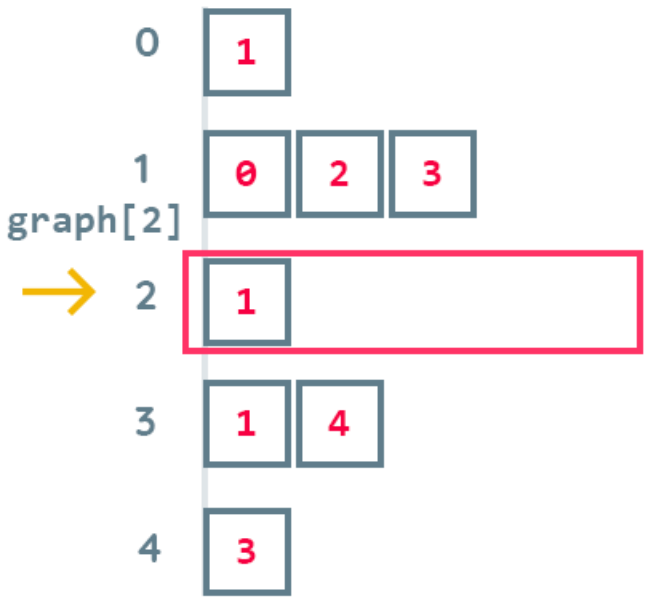
DFS Simulation




```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

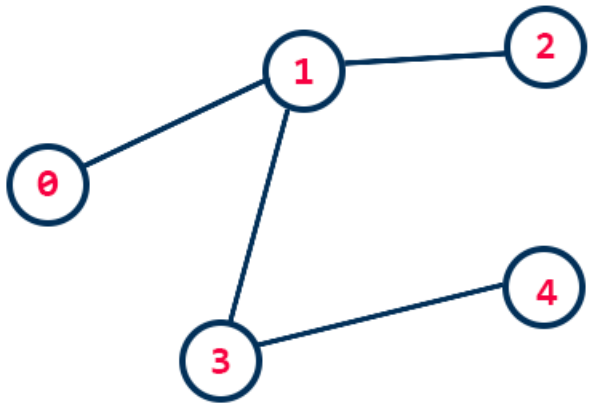


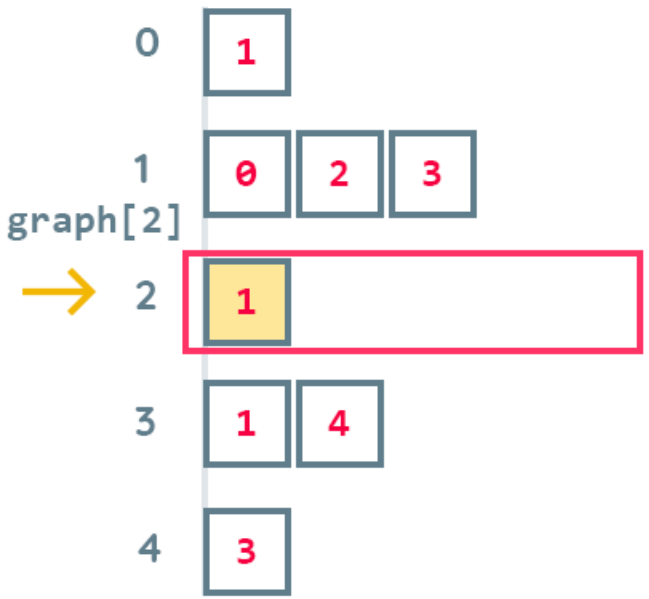
nodes: 0 1 2 3 4

source: 2

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

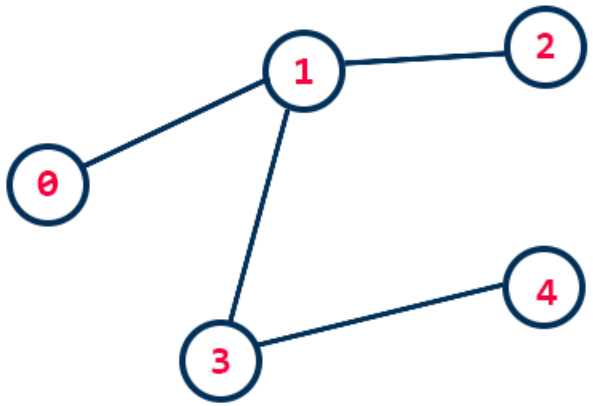
```
// 1 means that index(node)
// is visited
```

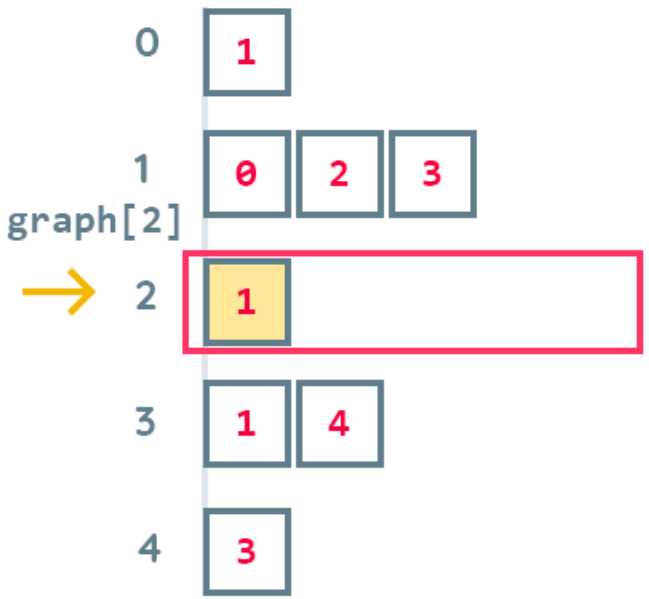
nodes: 0 1 2 3 4

source: 2

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

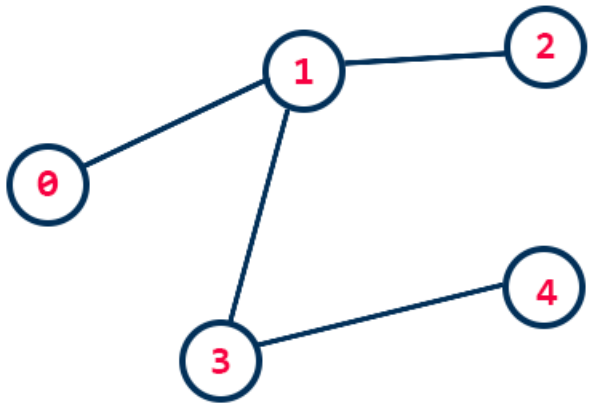
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

source: 2

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

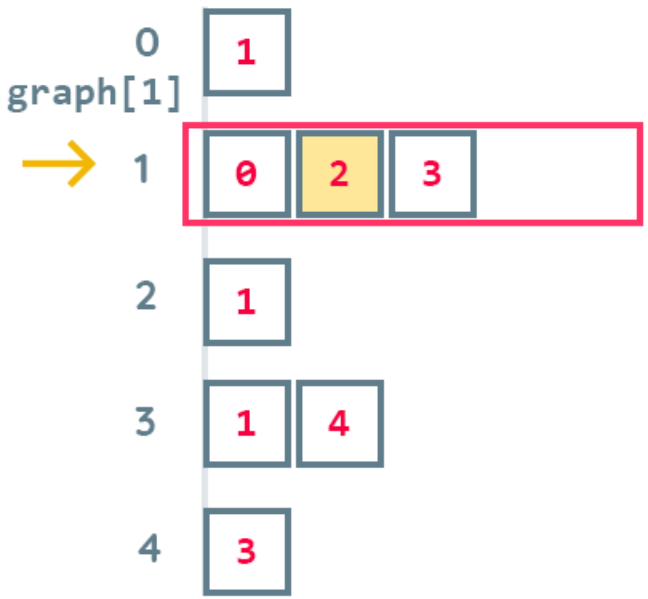
DFS Simulation




```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

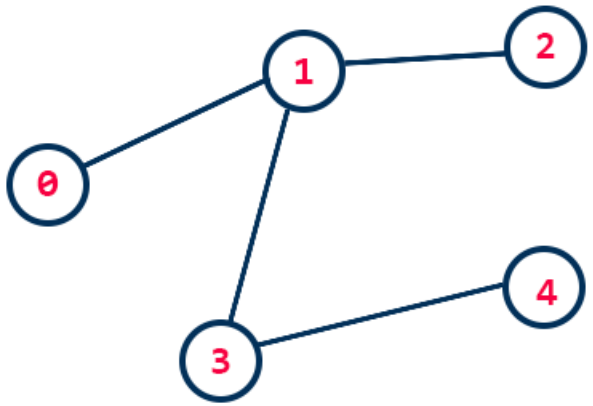


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

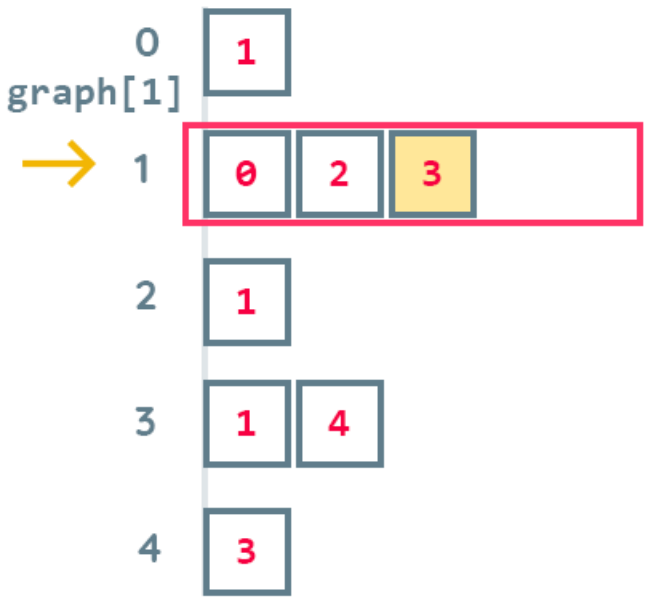
DFS Simulation




```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

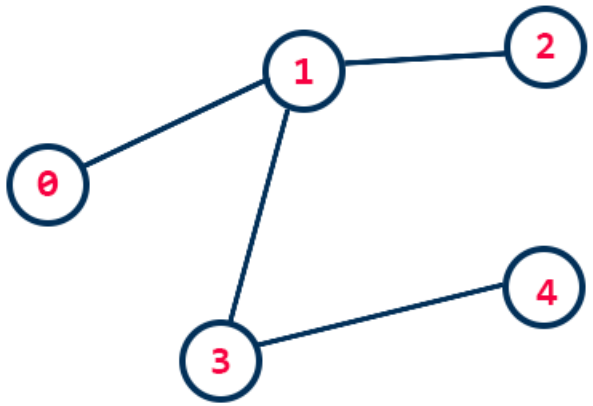


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

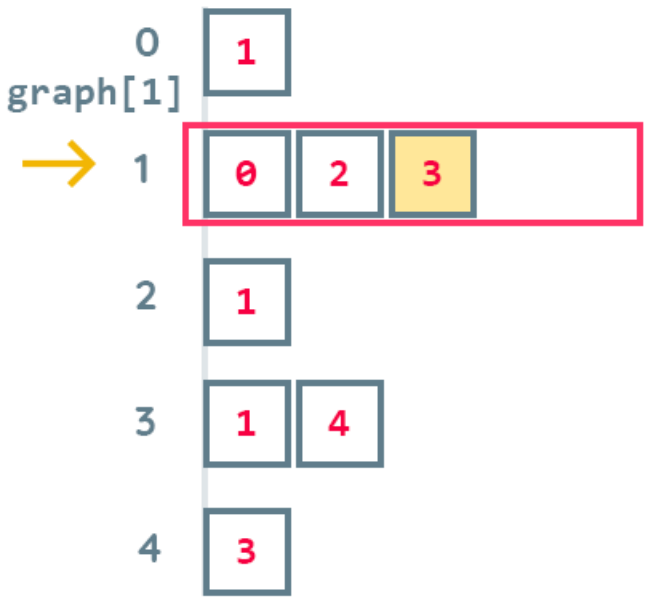
DFS Simulation




```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

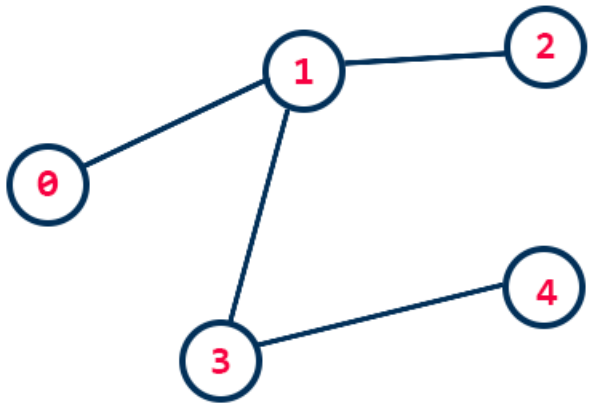


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

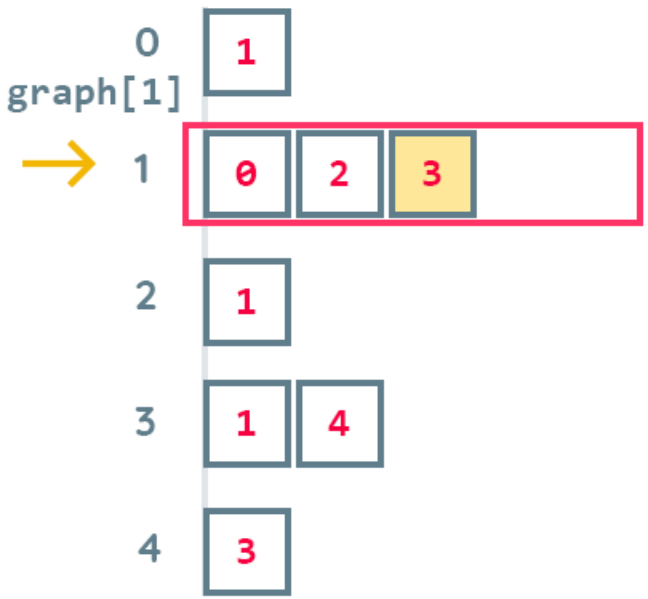
DFS Simulation




```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

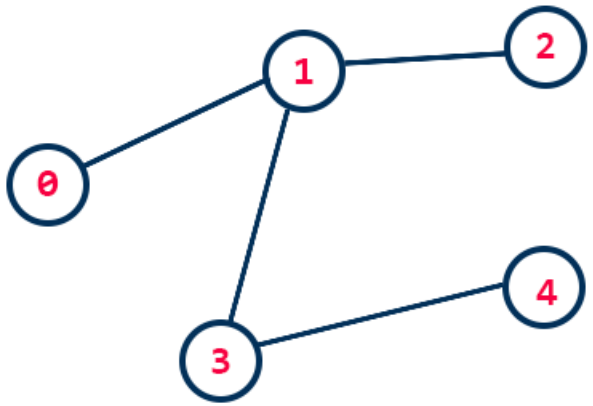


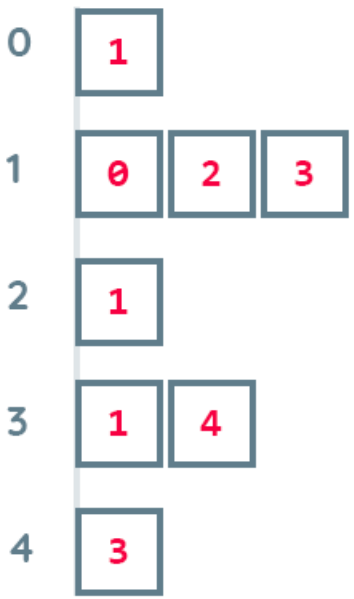
nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	0	0
0	1	2	3	4

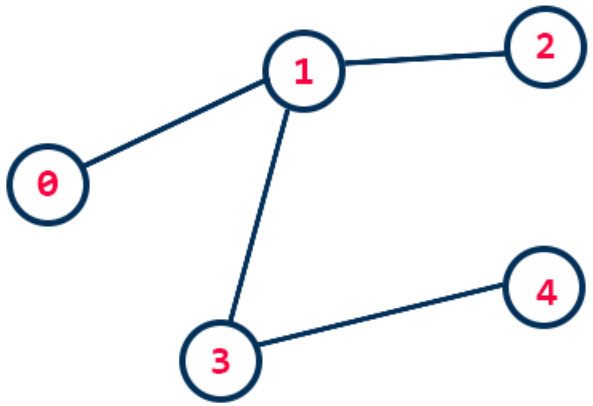
```
// 1 means that index(node)
// is visited
```

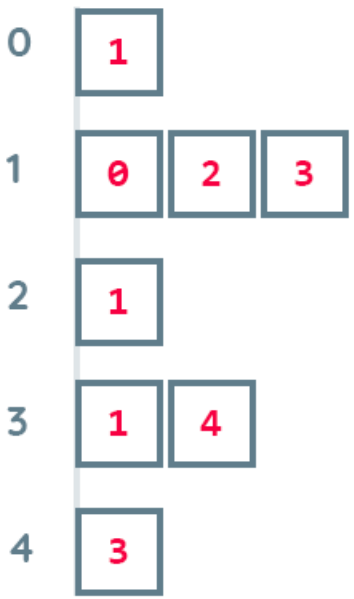
nodes: 0 1 2 3 4

source: 3

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	1	0
0	1	2	3	4

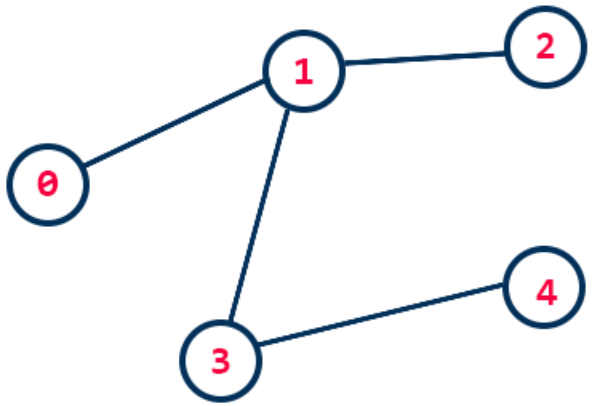
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

source: 3

```
dfs(source):
  visited[source] = 1
  for next in graph[source]:
    if not visited[next]:
      dfs(next)
```

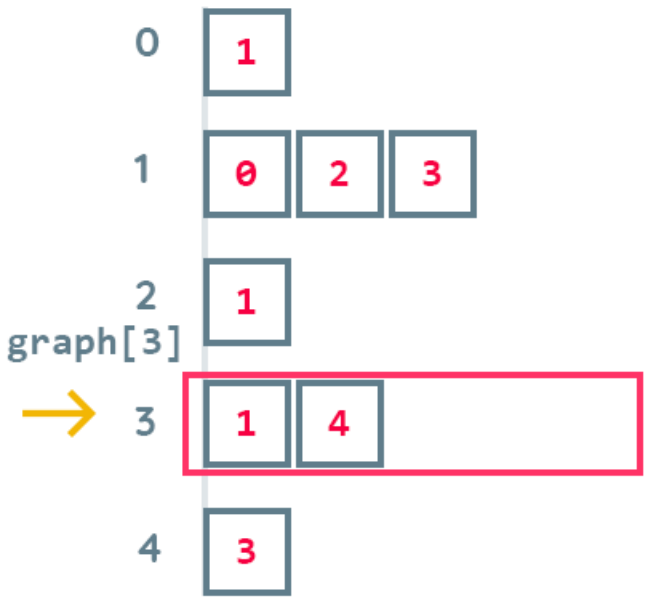
DFS Simulation




```
bool visited[5];
```

1	1	1	1	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

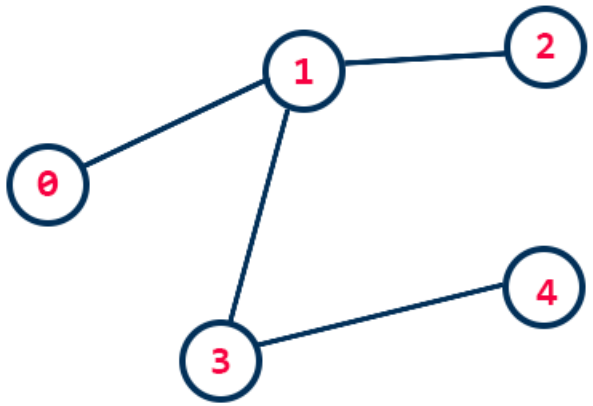


nodes: 0 1 2 3 4

source: 3

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

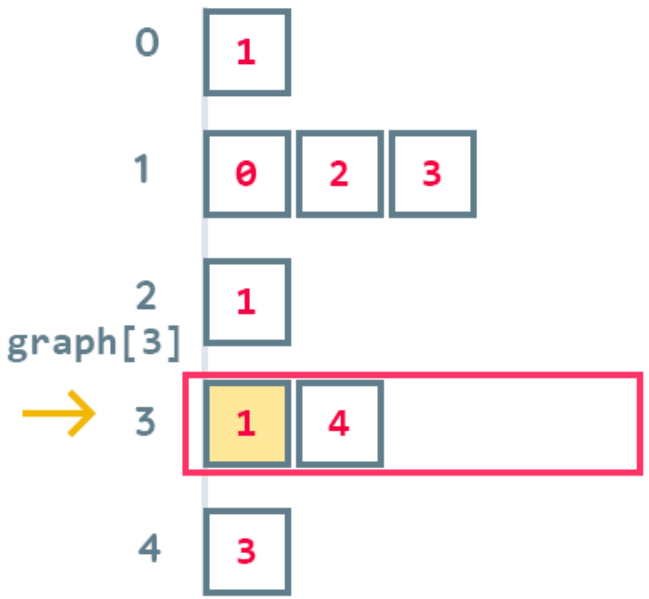
DFS Simulation




```
bool visited[5];
```

1	1	1	1	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

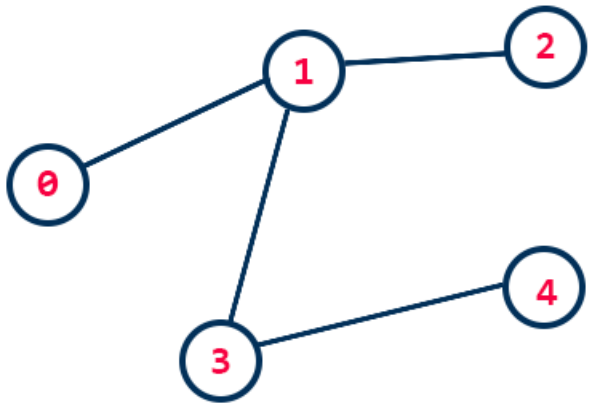


nodes: 0 1 2 3 4

source: 3

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

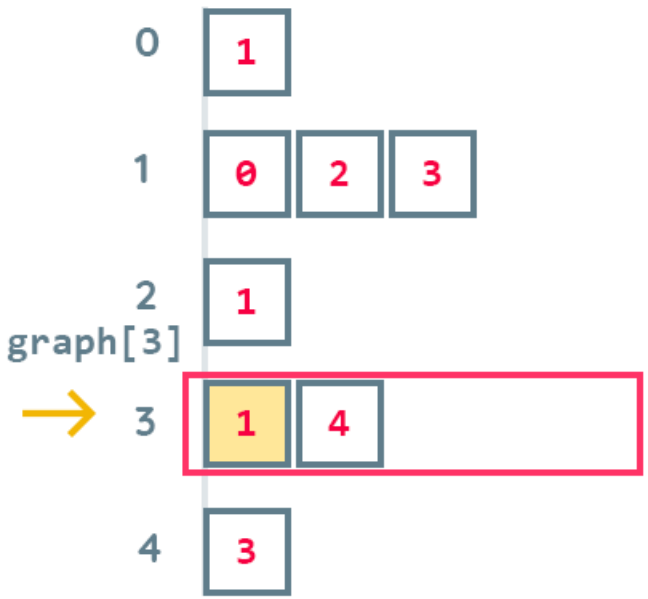
DFS Simulation




```
bool visited[5];
```

1	1	1	1	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

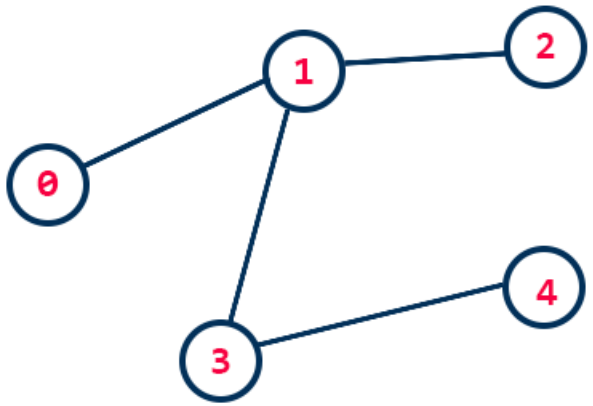


nodes: 0 1 2 3 4

source: 3

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

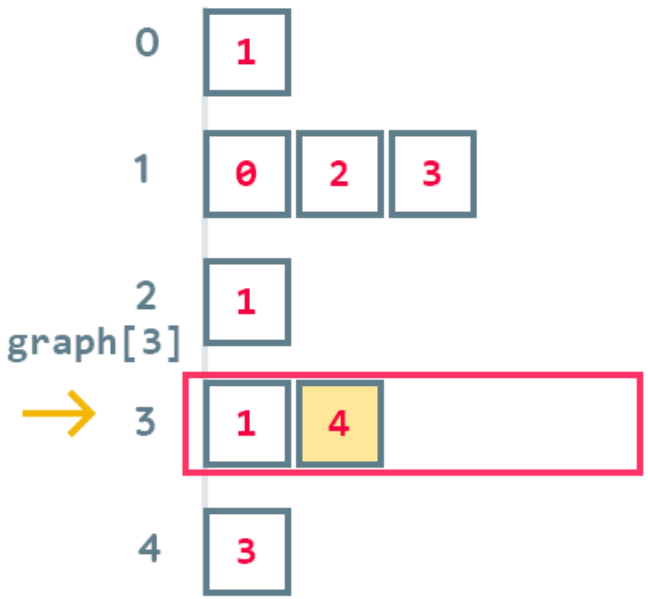
DFS Simulation




```
bool visited[5];
```

1	1	1	1	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

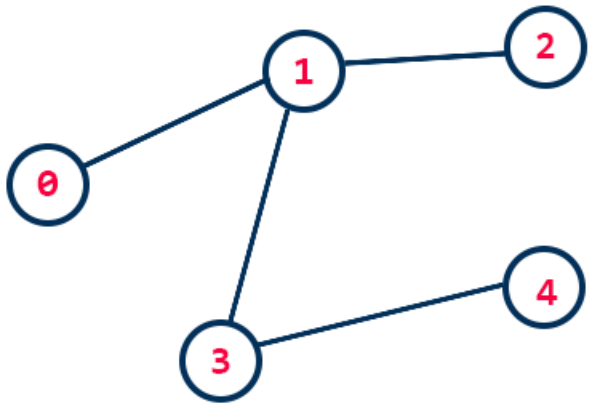


nodes: 0 1 2 3 4

source: 3

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

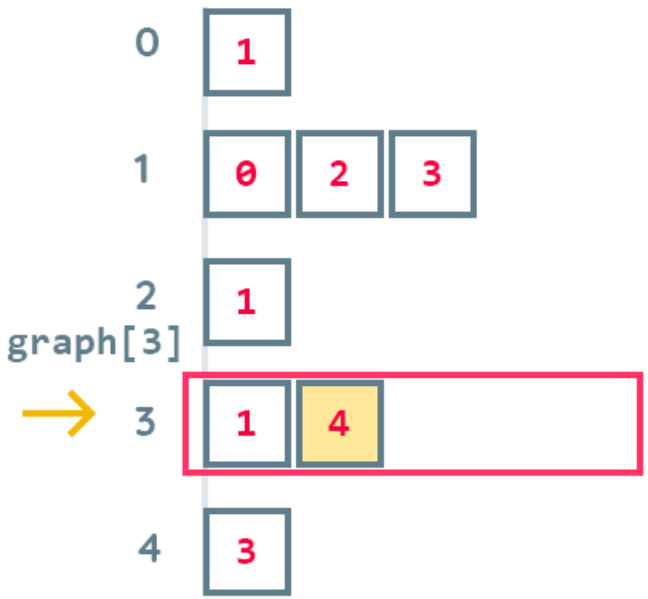
DFS Simulation




```
bool visited[5];
```

1	1	1	1	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

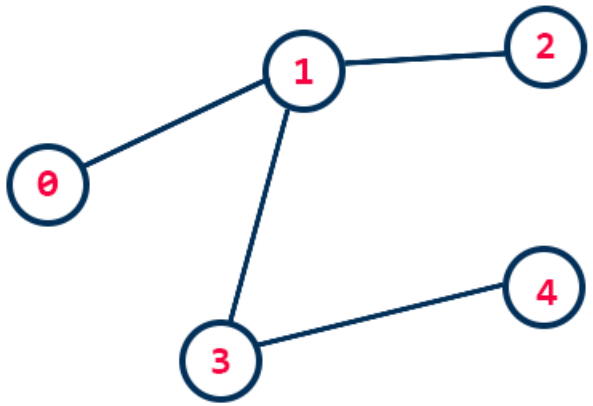


nodes: 0 1 2 3 4

source: 3

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

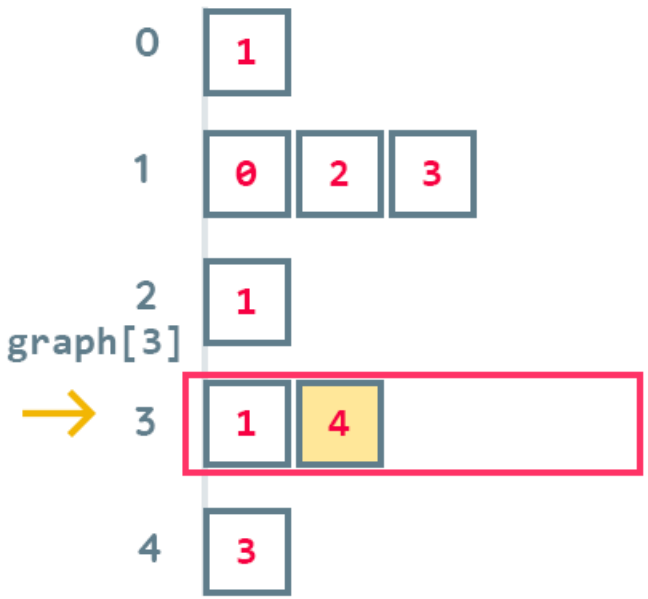
DFS Simulation




```
bool visited[5];
```

1	1	1	1	0
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

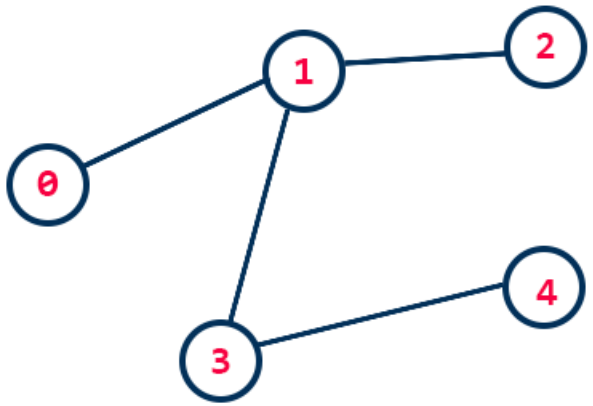


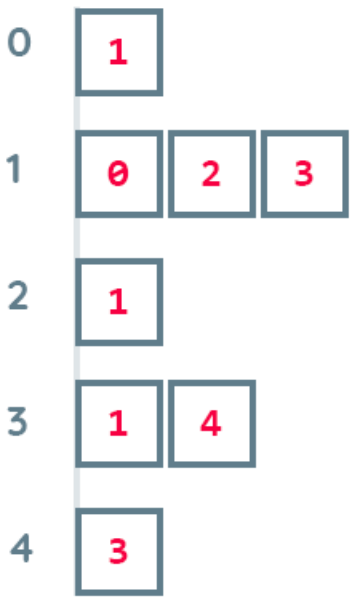
nodes: 0 1 2 3 4

source: 3

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	1	0
0	1	2	3	4

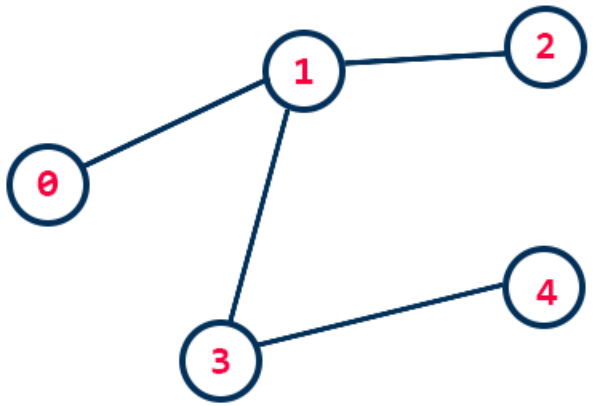
```
// 1 means that index(node)
// is visited
```

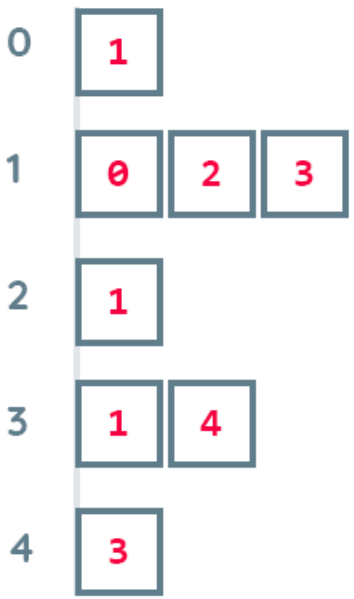
nodes: 0 1 2 3 4

source: 4

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

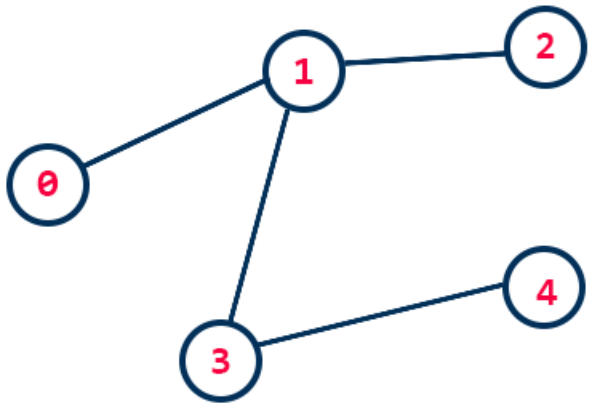
```
// 1 means that index(node)
// is visited
```

nodes: **0** 1 2 3 4

source: 4

```
dfs(source):
  visited[source] = 1
  for next in graph[source]:
    if not visited[next]:
      dfs(next)
```

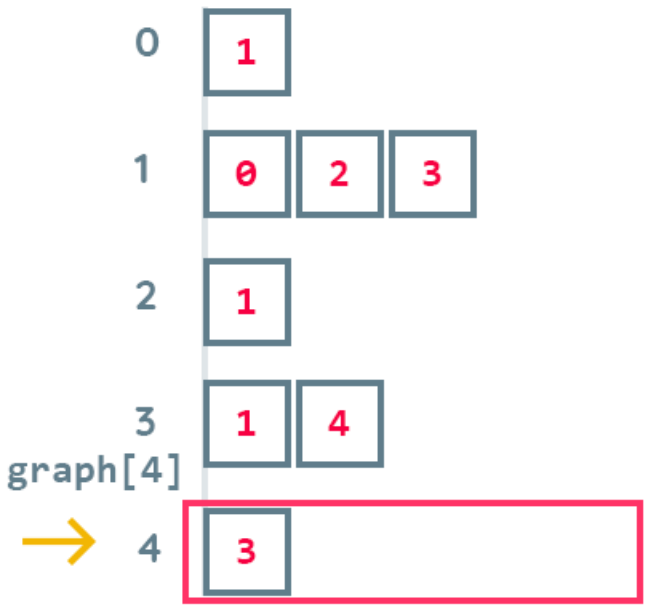
DFS Simulation




```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

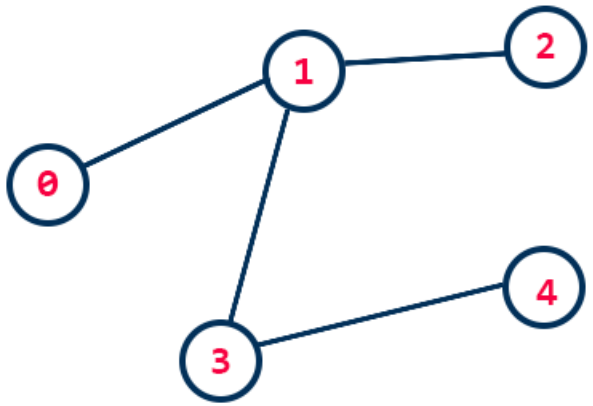


nodes: 0 1 2 3 4

source: 4

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

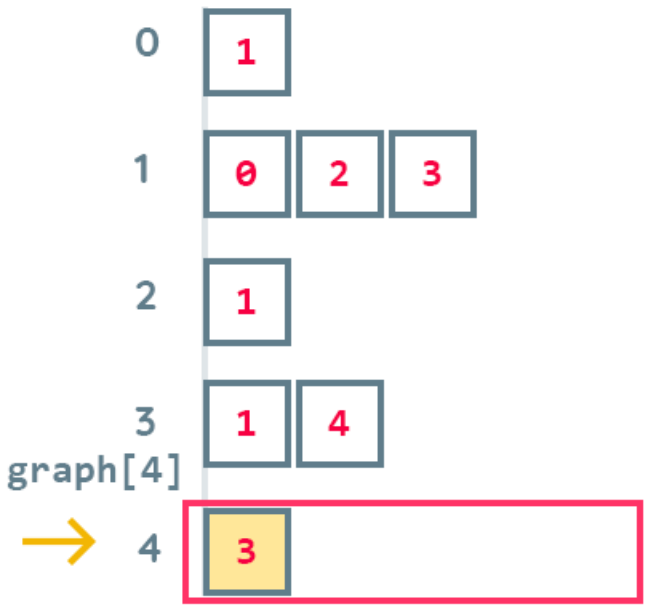
DFS Simulation




```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

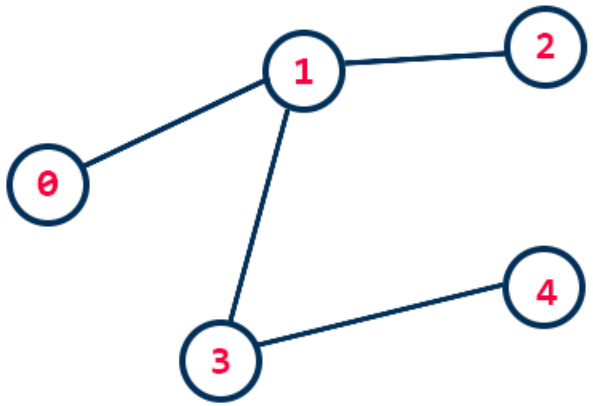


nodes: 0 1 2 3 4

source: 4

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

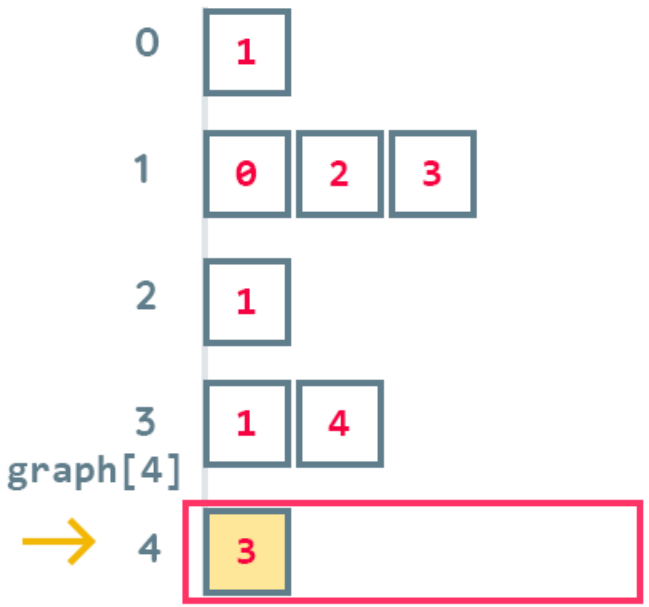
DFS Simulation




```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

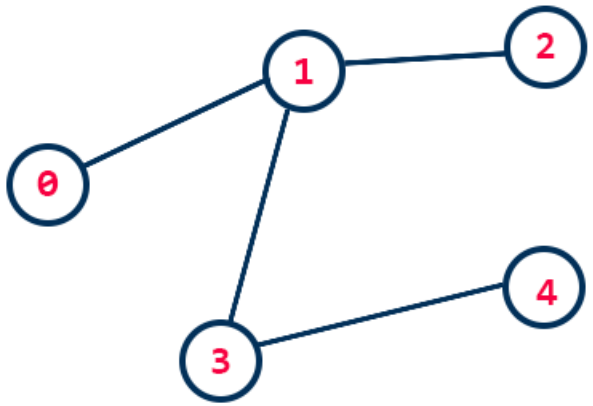


nodes: 0 1 2 3 4

source: 4

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

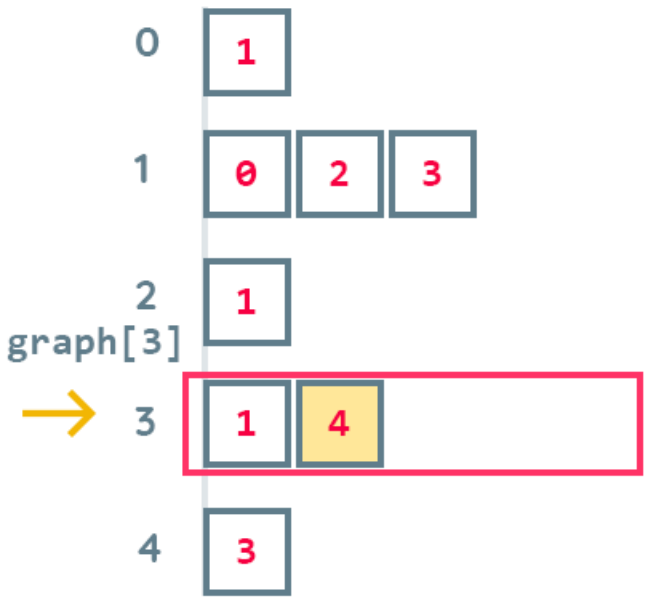
DFS Simulation




```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

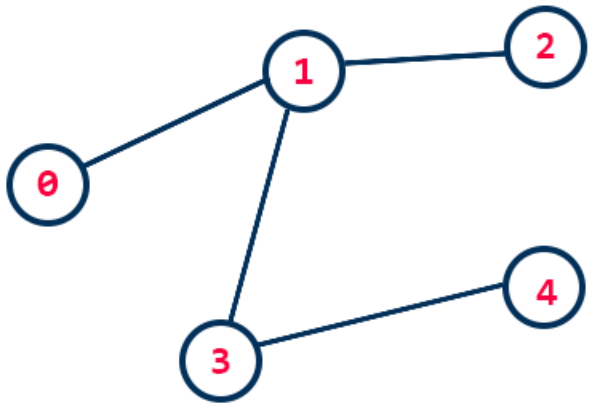


nodes: 0 1 2 3 4

source: 3

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

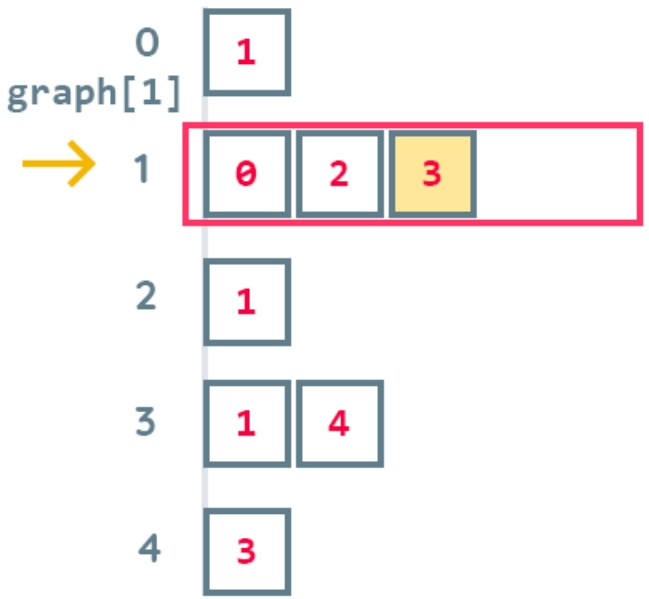
DFS Simulation




```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```

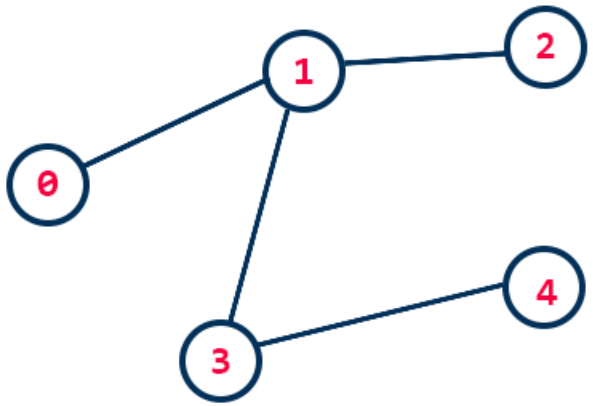


nodes: 0 1 2 3 4

source: 1

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation



graph[0]

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

```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

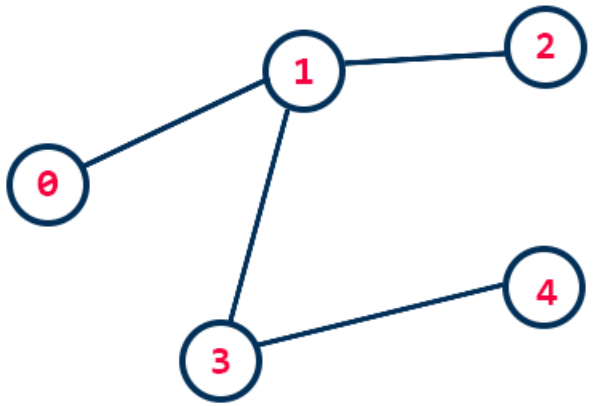
```
// 1 means that index(node)
// is visited
```

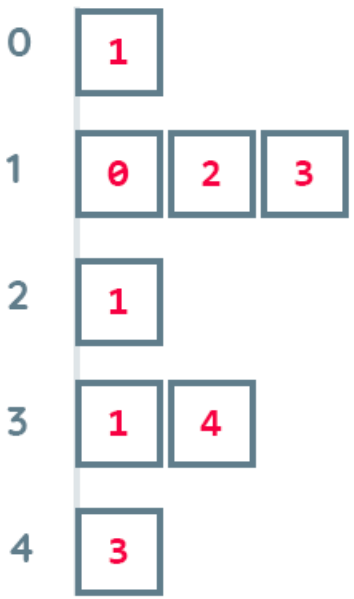
nodes: 0 1 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

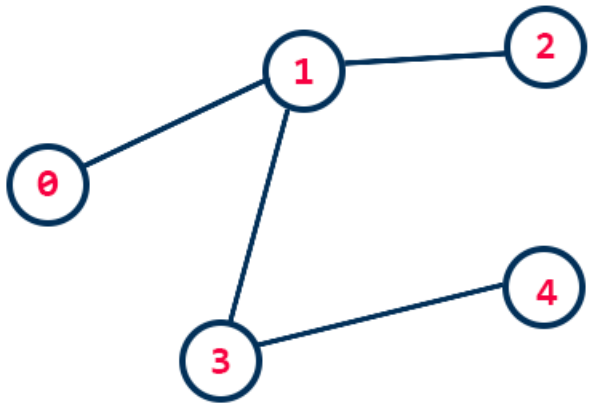
```
// 1 means that index(node)
// is visited
```

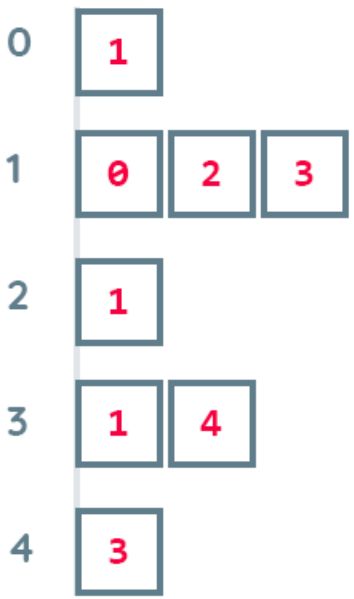
```
nodes: 0  1  2  3  4

source: 0
```

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

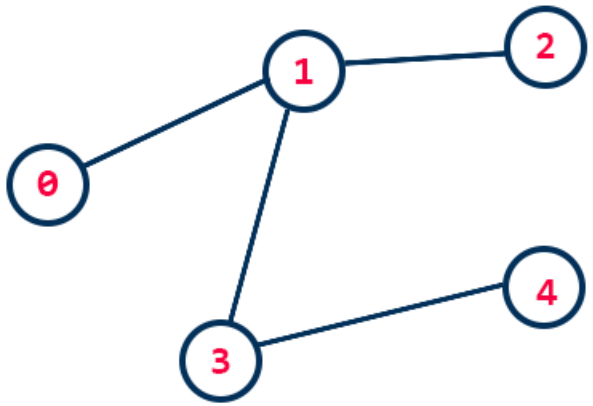
```
// 1 means that index(node)
// is visited
```

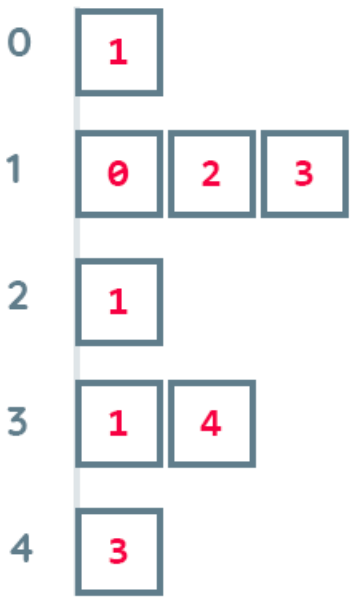
nodes: 0 **1** 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

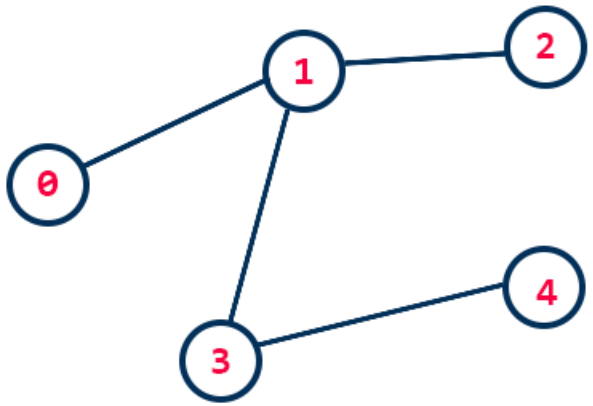
```
// 1 means that index(node)
// is visited
```

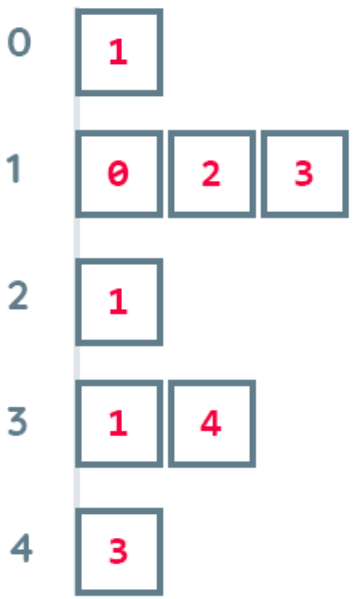
nodes: 0 1 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

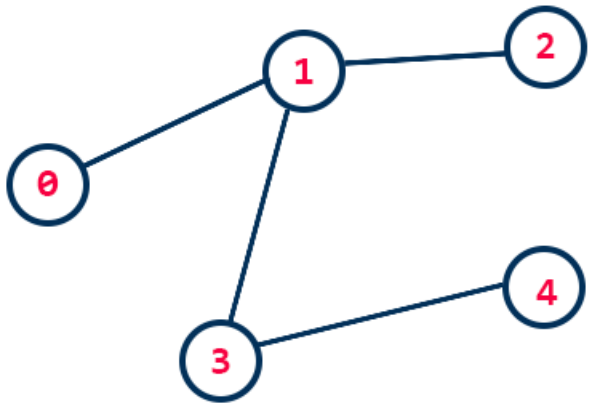
```
// 1 means that index(node)
// is visited
```

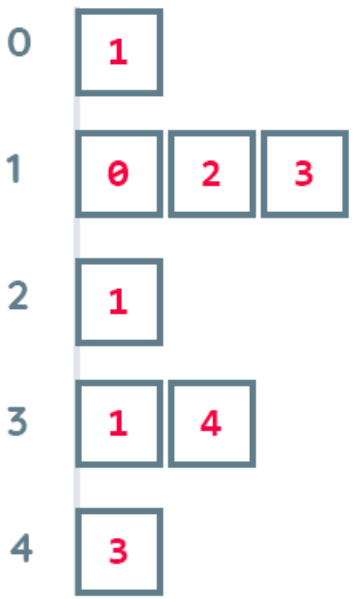
nodes: 0 1 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

DFS Simulation






```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

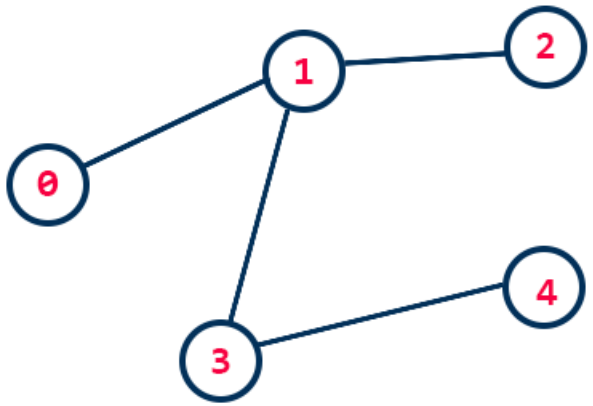
```
// 1 means that index(node)
// is visited
```

nodes: 0 1 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```

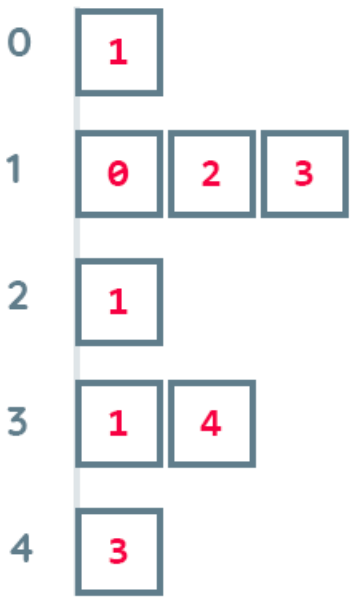
DFS Simulation




```
bool visited[5];
```

1	1	1	1	1
0	1	2	3	4

```
// 1 means that index(node)
// is visited
```



nodes: 0 1 2 3 4

source: 0

```
dfs(source):
   visited[source] = 1
   for next in graph[source]:
      if not visited[next]:
         dfs(next)
```