

DSA using Python

Graph traversing



Saurabh Shukla (MySirG)

Agenda

- ① Traversing Graph
- ② BFS
- ③ Logic for BFS
- ④ Example
- ⑤ DFS
- ⑥ Logic for DFS
- ⑦ Example

Traversing

There are two standard way of traversing a graph.

- ① BFS Breadth First Search
- ② DFS Depth First Search

BFS

Traversing graph has only issue that graph may have cycle. You may revisit a node.

To avoid processing a node more than once, we divide the vertices into two categories :

- ① visited
- ② Not visited

A boolean visited array is used to mark the visited vertices

BFS (Breadth First Search) uses a queue data structure for traversal.

Traversing begin from a node called source node.

BFS(G, S)

Logic for BFS

Let Q be the queue

Q.insert(S)

v[S] = True

while (!Q.isEmpty())

{ n = Q.getFront()

Q.del()

for all the neighbors u of n

if v[u] == False

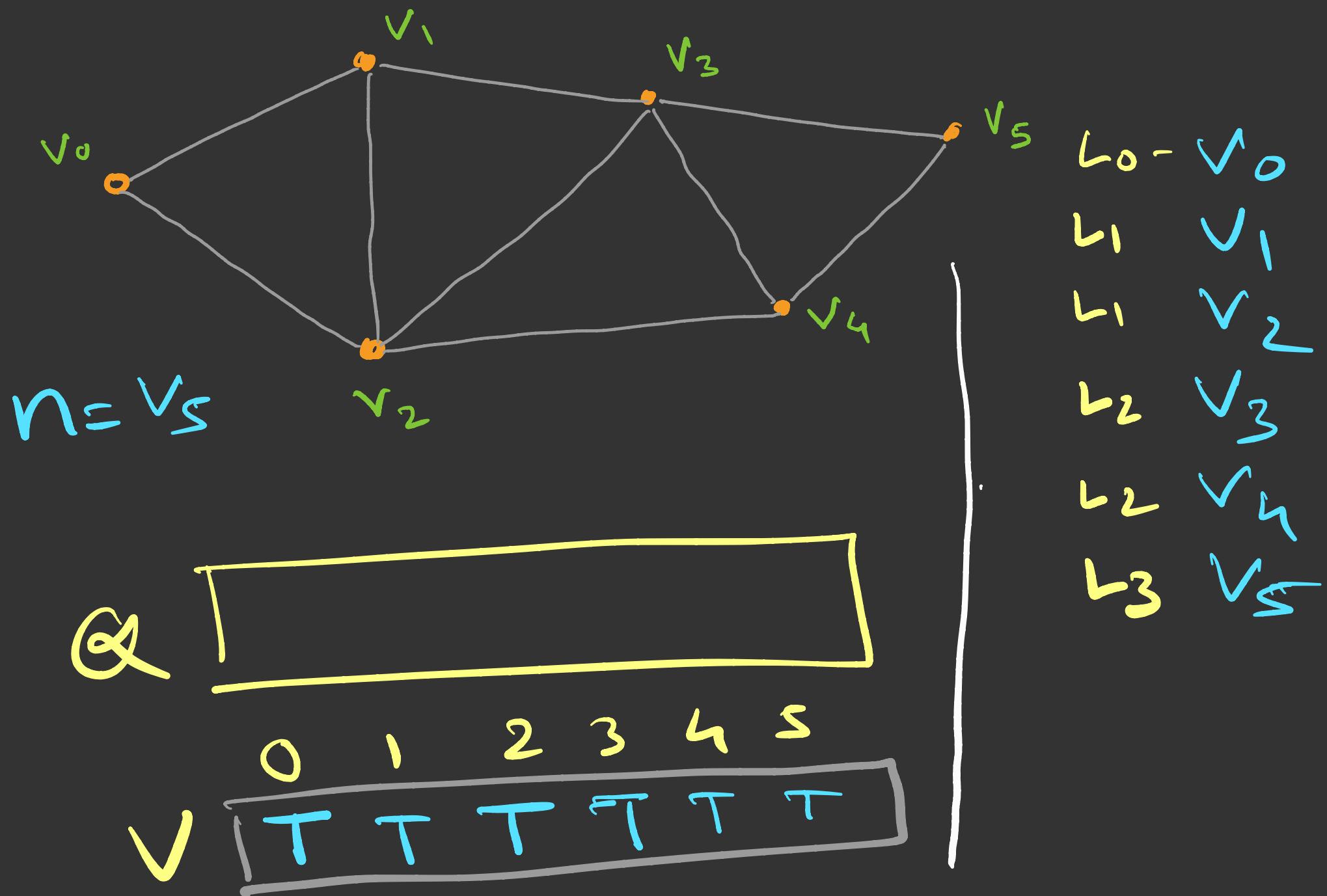
Q.insert(u)

v[u] = True

}

Source = v_0

Example



DFS

DFS is Depth first search.

The main difference between BFS and DFS is that the DFS uses stack in place of Queue.

DFS (G, s)

Logic for DFS

Let stack be the stack

stack.push(s)

$v[s] = \text{True}$

while (!stack.isEmpty())

{

$n = \text{stack.peek()}$

stack.pop()

for all the neighbours u of n

if $v[u] == \text{False}$

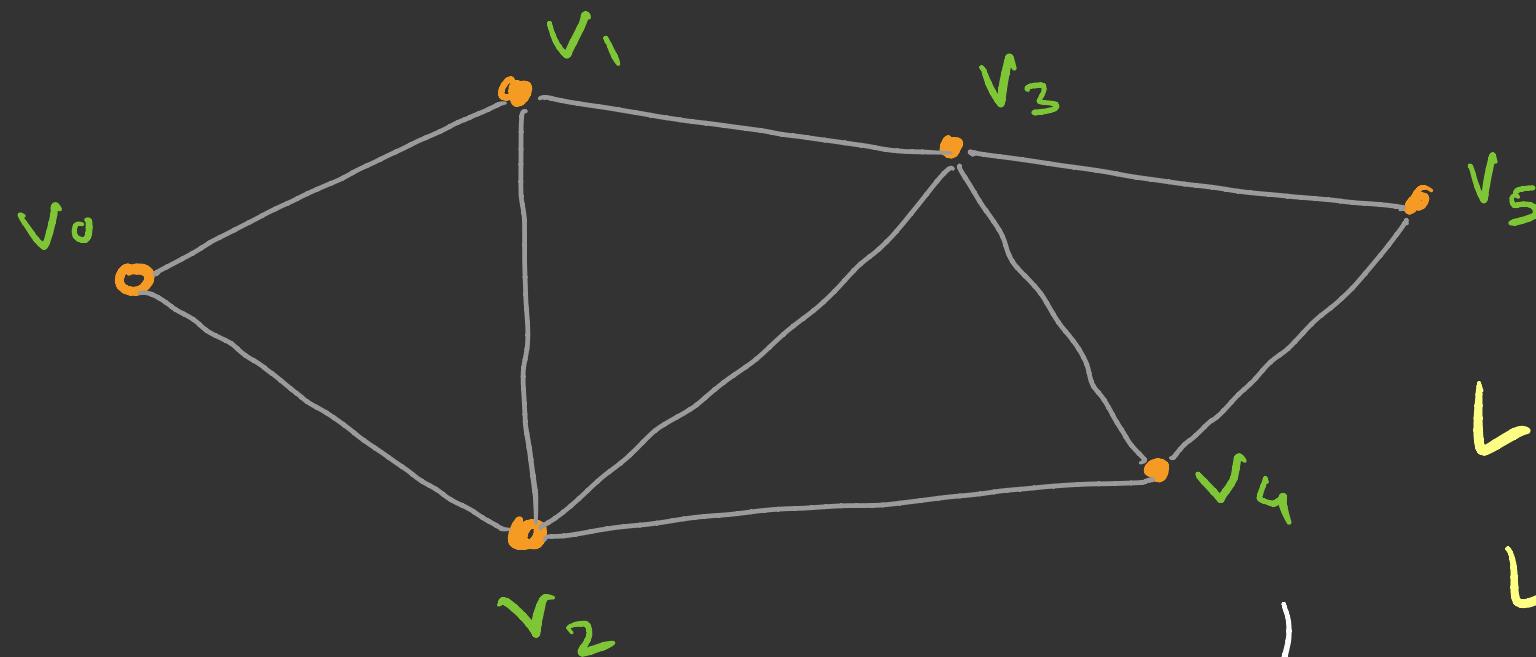
stack.push(u)

$v[u] = \text{True}$

}

Source = v_0

Example



$$n = v_1$$

