BFS Algorithm code

#!/usr/bin/env python

# coding: utf-8

# In[ ]:

from collections import defultdict

from queue import Queue

# In[ ]:

class Graph():

def\_init\_(self,directed):

    self.graph = defaultdict(list)

    self.directed =directed

    def add\_edge(self,u,v):

        if  self.directed:

        self.graph[u].append(v)

        else:

        self.graph[u].aparend(v)

        self.graph[v].aparend(u)

        def bfs(self, vertex):

            visited =[]

        queue = Queue()

        queue.put(vertex)

        while not queue empty():

            vertex=queue.get()

            if vertex in visited:

                continue

                print(vertex,end =" ")

                visited.aparend(vertex)

                for neighbour in self.graph[vertex]:

                    if neighbour != None:

                    queue.put(neighbour)

# In[ ]:

g = Graph(True)

# In[4]:

g.add\_edge('s','r')

g.add\_edge('s','v')

g.add\_edge('s','x')

g.add\_edge('r','t')

g.add\_edge('v','w')

g.add\_edge('x','r')

g.add\_edge('x','u')

g.add\_edge('t','x')

g.add\_edge('t','u')

g.add\_edge('t','y')

g.add\_edge('w','s')

g.add\_edge('w','y')

g.add\_edge('u',None)

g.add\_edge('y','u')

# In[5]:

g.graph

# In[1]:

g.bfs('s')

Output :

srvxtwuy