Best First Search(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 of the code :

```
srvxtwuy
```