```python
import networkx as nx
import matplotlib.pyplot as plt


def visita_profondita(grafo, vertice_start, plot_grafo=False, plot_albero=False, plot_anim=False):
    """
    Visita in profondita' di un grafo con albero di ricoprimento

    :param grafo: deve essere un grafo di networkx
    :param vertice_start: nodo arbitrario del grafo
    :param plot_grafo: booleano che indica se visualizzare il grafo da visitare
    :param plot_albero: booleano che indica se visualizzare l'albero di ricoprimento del grafo
    :param plot_anim: booleano che indica se visualizzare l'animazione della visita del grafo
    :return: albero di ricoprimento della visita
    """

    albero = nx.Graph()
    nodi_da_visitare = []
    nodi_da_visitare.append(vertice_start)

    pos = nx.spring_layout(grafo)

    # Le due liste "nodi_scoperti" e "nodi_non_scoperti" servono per l'animazione della visita
    if plot_anim is True:
        nodi_scoperti = [vertice_start]
        nodi_non_scoperti = list(grafo.nodes())
        nodi_non_scoperti.remove(vertice_start)

    for v in grafo.nodes():
        if v != vertice_start:
            grafo.nodes[v]['colore'] = 'blu'  # blu -> non scoperto

    grafo.nodes[vertice_start]['colore'] = 'verde'  # verde -> scoperto
    albero.add_node(vertice_start)

    plt.axes().clear()
    while len(nodi_da_visitare) > 0:

        v_green = nodi_da_visitare.pop()  # la mia struttura dati e' una pila: rimuovo l'ultimo elemento inserito

        for v in list(grafo[v_green]):

            if plot_anim is True:
                nx.draw_networkx(grafo, pos, nodelist=nodi_non_scoperti, node_color='b')
                nx.draw_networkx(grafo, pos, nodelist=nodi_scoperti, node_color='g')
                plt.pause(0.2)
                plt.axes().clear()

            if grafo.nodes[v]['colore'] != 'verde':
                nodi_da_visitare.append(v)
                grafo.nodes[v]['colore'] = 'verde'
                albero.add_node(v)
                albero.add_edge(v_green, v)
                if plot_anim is True:
                    nodi_non_scoperti.remove(v)
                    nodi_scoperti.append(v)

    if plot_grafo is True:
        nx.draw_networkx(grafo, pos)
        plt.pause(2)
        plt.axes().clear()

    if plot_albero is True:
        nx.draw_networkx(albero, pos)
```

```
65          plt.pause(2)
66          plt.axes().clear()
67
68      return albero
69
70
71  def visita_ampiezza(grafo, vertice_start, plot_grafo=False, plot_albero=False, plot_anim
    =False):
72      """
73      Visita in ampiezza di un grafo con albero di ricoprimento
74
75      :param grafo: deve essere un grafo di networkx
76      :param vertice_start: nodo arbitrario del grafo
77      :param plot_grafo: booleano che indica se visualizzare il grafo da visitare
78      :param plot_albero: booleano che indica se visualizzare l'albero di ricoprimento del
     grafo
79      :param plot_anim: booleano che indica se visualizzare l'animazione della visita del
    grafo
80      :return: albero di ricoprimento della visita
81      """
82
83      albero = nx.Graph()
84      nodi_da_visitare = []
85      nodi_da_visitare.append(vertice_start)
86
87      pos = nx.spring_layout(grafo)
88
89      # Le due liste "nodi_scoperti" e "nodi_non_scoperti" servono per l'animazione della
    visita
90      if plot_anim is True:
91          nodi_scoperti = [vertice_start]
92          nodi_non_scoperti = list(grafo.nodes())
93          nodi_non_scoperti.remove(vertice_start)
94
95      for v in grafo.nodes():
96          if v != vertice_start:
97              grafo.nodes[v]['colore'] = 'blu'  # blu -> non scoperto
98
99      grafo.nodes[vertice_start]['colore'] = 'verde'  # verde -> scoperto
100     albero.add_node(vertice_start)
101
102     plt.axes().clear()
103     while len(nodi_da_visitare) > 0:
104
105         v_green = nodi_da_visitare.pop(0)  # la mia struttura dati e' una coda: rimuovo
    il primo elemento inserito
106
107         for v in list(grafo[v_green]):
108
109             if plot_anim is True:
110                 nx.draw_networkx(grafo, pos, nodelist=nodi_non_scoperti, node_color='b')
111                 nx.draw_networkx(grafo, pos, nodelist=nodi_scoperti, node_color='g')
112                 plt.pause(0.2)
113                 plt.axes().clear()
114
115             if grafo.nodes[v]['colore'] != 'verde':
116                 nodi_da_visitare.append(v)
117                 grafo.nodes[v]['colore'] = 'verde'
118                 albero.add_node(v)
119                 albero.add_edge(v_green, v)
120                 if plot_anim is True:
121                     nodi_non_scoperti.remove(v)
122                     nodi_scoperti.append(v)
123
124     if plot_grafo is True:
125         nx.draw_networkx(grafo, pos)
126         plt.pause(2)
127         plt.axes().clear()
128
```

```
129     if plot_albero is True:
130         nx.draw_networkx(albero, pos)
131         plt.pause(2)
132         plt.axes().clear()
133 
134     return albero
135 
```