Prim’s Algorithm

def prims\_algorithm(self, starting\_node):  
 # we keep an array to know if a vertex was selected or not, in order to avoid cycles  
 selected = [False] \* self.\_\_vertices\_number  
 edge\_number = 0  
 selected[starting\_node] = True  
 print(" Edge | Weight")  
 while edge\_number < self.\_\_vertices\_number - 1:  
 # the algorithm ends when we reach V - 1 nodes in the tree  
 minimum = sys.maxsize  
 source\_node = 0  
 destination\_node = 0  
 for i in range(len(selected)):  
 if selected[i]:  
 for j in range(len(selected)):  
 # if the node hasn't been added to the tree yet and there is an edge with the selected node,  
 # we check to see if the edge has the minimum possible cost  
 if not(selected[j]) and self.\_\_graph\_matrix[i][j]:  
 if minimum > self.\_\_graph\_matrix[i][j]:  
 minimum = self.\_\_graph\_matrix[i][j]  
 source\_node = i  
 destination\_node = j  
 print(str(source\_node) + " - " + str(destination\_node) + " | " + str(self.\_\_graph\_matrix[source\_node][destination\_node]))  
 selected[destination\_node] = True  
 edge\_number += 1