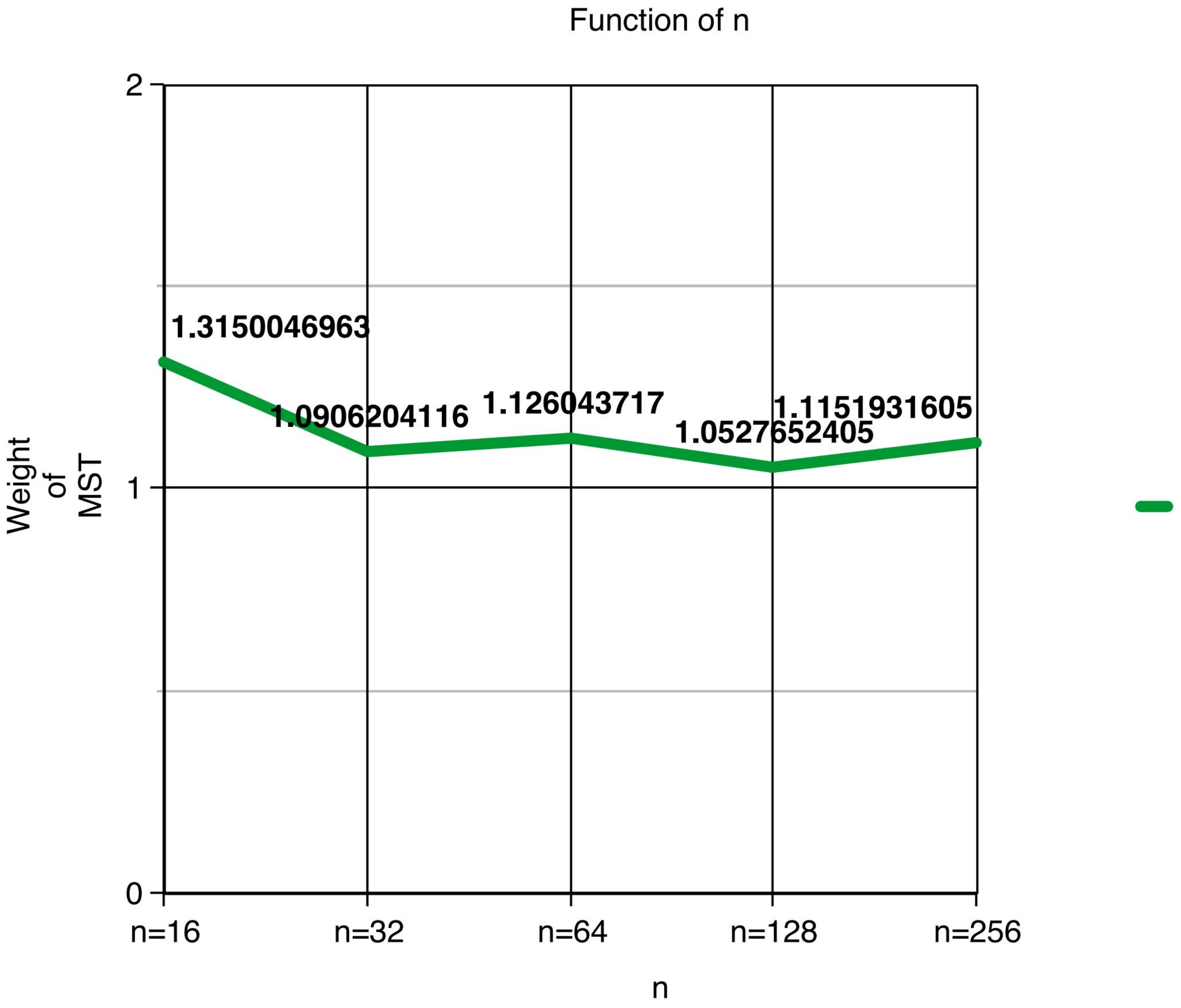
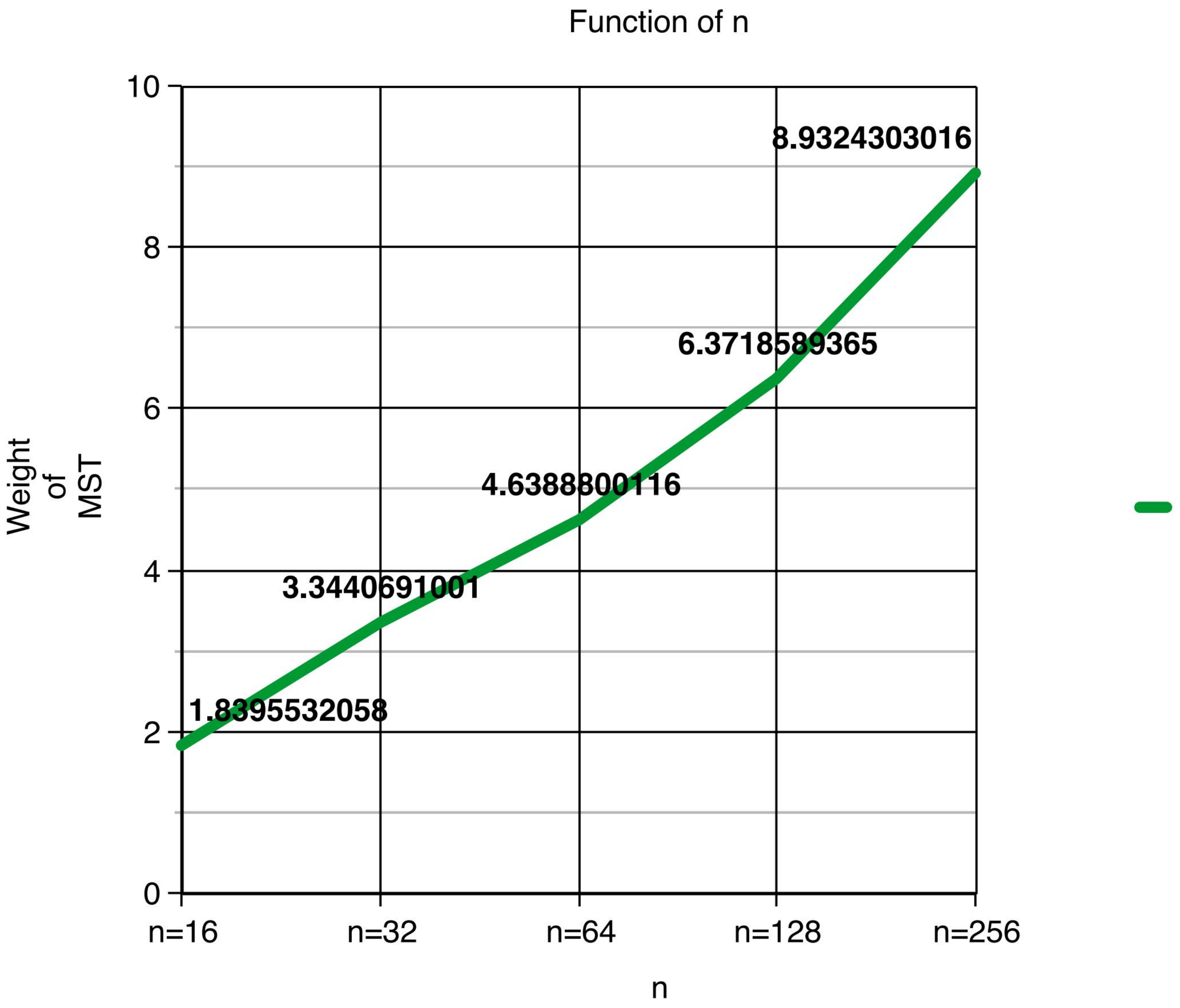
```
import random


import math


n = 256

number = []
# the edge weight


position = []
#the 2 vertices of an edge


for i in range(n):
            for j in range(i):
                        position += [[j,i]]
   #                     number += [random.uniform(0,1)]
                        # case for part a

                        number += [math.sqrt((random.uniform(0,1)-
random.uniform(0,1))**2 +(random.uniform(0,1)-
random.uniform(0,1))**2)]

                        #case for part b.



#initilize positions and weights


m = [[0 for x in range(n)] for y in range(n)]

for i in range(len(position)):
            m[position[i][0]][position[i][1]] = number[i]
            m[position[i][1]][position[i][0]] = number[i]
for i in range(len(m)):
            for j in range(len(m[i])):
                        m[i][j] = [m[i][j],i,j]

# initializing the lists


#print(m)
# print edge list



us = []
#used list   helper list

uu = []
```

```
#unused list   initially contains the every vertex


edge_list = []
#edge list

MST = []
#Minimum Spanning tree

for i in range(n):
            uu += [i]
            #initilizing by putting every node into the unused list

while (uu != []):
    #while unused list is not empty

            if (us == []):
                        random_number = random.randint(0,len(uu)-1)
                        us += [random_number]
                        uu = uu[:random_number]+uu[random_number+1:]
                        edge_list += m[random_number]
[:random_number] + m[random_number][random_number+1:]
            # initializing the used list at first

            #print(us)
            #print(uu)

            a = min(edge_list)
            # find the minimum weight neighbors

            MST += [a[0]]
            #add into our MST
            c = [a[0],a[2],a[1]]
            # the case where node 1 to node 2 and node 2 to node 1
is repeated
            us += [a[2]]
            uu.remove(a[2])
            edge_list += m[a[2]][:a[2]]+m[a[2]][a[2]+1:]
            edge_list.remove(a)
            edge_list.remove(c)
            # remove repetitive case

            edge_list2 = list(edge_list)

            for i in range(len(edge_list2)):
                        #delete the case and nodes we dont want in
the edge list
                        if edge_list2[i][2] in us:
                                    edge_list.remove(edge_list2[i])


print(sum(MST))
#calculate the weight of MST
```


Function of n
2
1
0
Weight of MST
1.3150046963
1.0906204116
1.126043717
1.0527652405
1.1151931605
n=16
n=32
n=64
n=128
n=256
n


Function of n
Weight of MST
10
8
6
4
2
0
n=16
n=32
n=64
n=128
n=256
n
1.8395532058
3.3440691001
4.6388800116
6.3718589365
8.9324303016