Krish Mehta

D12A / 42

AI Lab: Exp 9

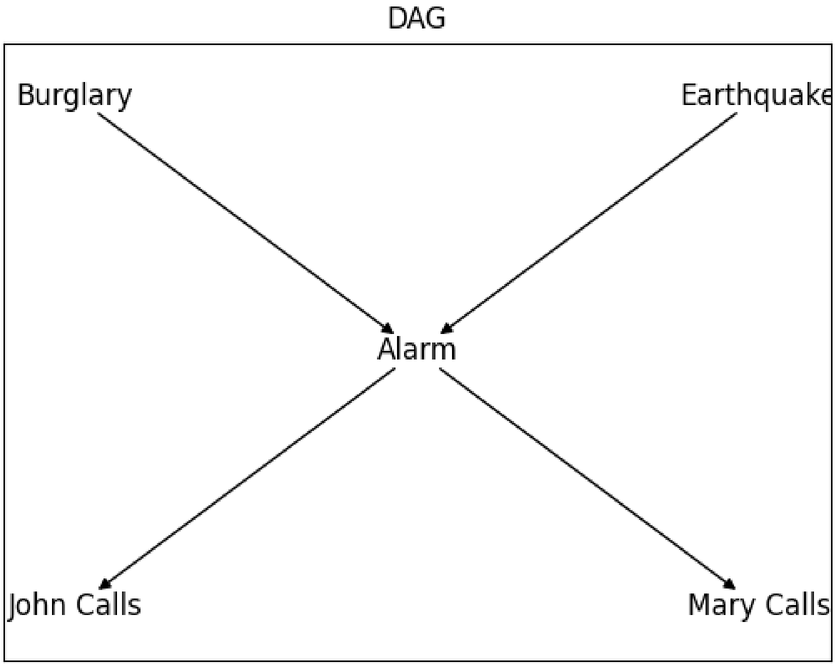
Aim: To implement Directed Acyclic Graph (DAG) for the given problem statement using python.

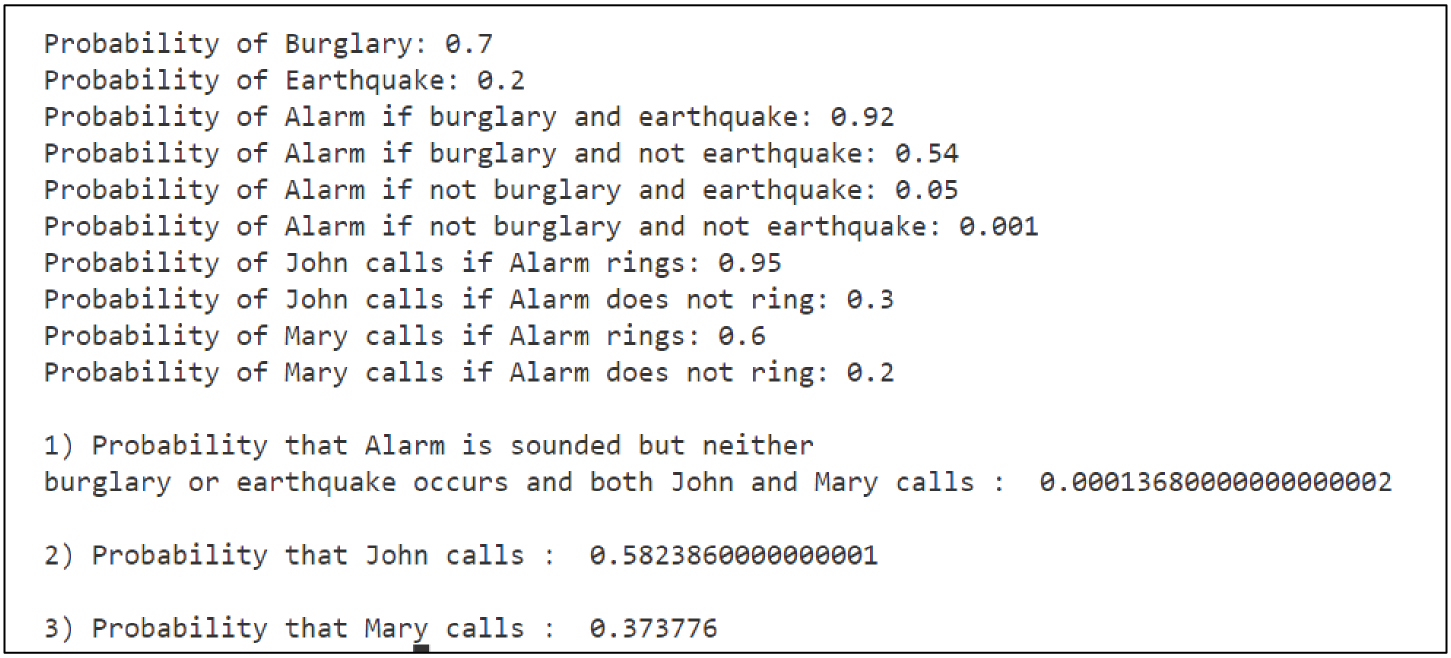
Code:

```
from matplotlib import pyplot as plt
import networkx as nx
import numpy as np

#Graph
G = nx.DiGraph()
nodes = np.arange(0, 5).tolist()
G.add_nodes_from(nodes)
G.add_edges_from([(0,2), (1,2), (2,3), (2,4)])
pos = {0:(5, 10), 1:(10, 10), 2:(7.5, 7.5), 3: (5,5), 4: (10,5)}
labels ={0:"Burglary", 1:"Earthquake", 2:"Alarm", 3:"John Calls", 4:"Mary Calls"}
nx.draw_networkx(G, pos = pos, labels = labels, arrows = True, node_shape = "s",
                                                            node_color = "white")
plt.title("DAG")
plt.show()

#Solution
prob_b = float(input("Probability of Burglary: "))
prob_e = float(input("Probability of Earthquake: "))
prob_a_tt = float(input("Probability of Alarm if burglary and earthquake: "))
prob_a_tf = float(input("Probability of Alarm if burglary and not earthquake: "))
prob_a_ft = float(input("Probability of Alarm if not burglary and earthquake: "))
prob_a_ff = float(input("Probability of Alarm if not burglary and not erthqk: "))
prob_j_t = float(input("Probability of John calls if Alarm rings: "))
prob_j_f = float(input("Probability of John calls if Alarm does not ring: "))
prob_m_t = float(input("Probability of Mary calls if Alarm rings: "))
prob_m_f = float(input("Probability of Mary calls if Alarm does not ring: "))

ans1 = prob_j_t * prob_m_t * prob_a_ff * (1 - prob_b) * (1 - prob_e)
print("\n1) Probability that Alarm is sounded but neither",
      "\nburglary or earthquake occurs and both John and Mary calls : ", ans1)

prob_a = prob_a_tt * prob_b * prob_e + prob_a_tf * prob_b * (1 - prob_e) +
prob_a_ft * (1 - prob_b) * prob_e + prob_a_ff * (1 - prob_b) * (1- prob_e)

prob_not_a = (1 - prob_a_tt) * prob_b * prob_e + (1 - prob_a_tf) * prob_b * (1 -
prob_e) + (1 - prob_a_ft) * (1 - prob_b) * prob_e + (1 - prob_a_ff) * (1 -
prob_b) * (1- prob_e)
```

```
ans2 = (prob_j_t * prob_a) + (prob_j_f * prob_not_a)
print("\n2) Probability that John calls : ", ans2)
ans3 = (prob_m_t * prob_a) + (prob_m_f * prob_not_a)
print("\n3) Probability that Mary calls : ", ans3)
```

Output:





```
Probability of Burglary: 0.7
Probability of Earthquake: 0.2
Probability of Alarm if burglary and earthquake: 0.92
Probability of Alarm if burglary and not earthquake: 0.54
Probability of Alarm if not burglary and earthquake: 0.05
Probability of Alarm if not burglary and not earthquake: 0.001
Probability of John calls if Alarm rings: 0.95
Probability of John calls if Alarm does not ring: 0.3
Probability of Mary calls if Alarm rings: 0.6
Probability of Mary calls if Alarm does not ring: 0.2

1) Probability that Alarm is sounded but neither
burglary or earthquake occurs and both John and Mary calls :  0.00013680000000000002

2) Probability that John calls :  0.5823860000000001

3) Probability that Mary calls :  0.373776
```