



**Task 1:** convert the graph to Laplacian matrix (in degree), do this either on paper or just write out the matrix here

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Task 2 & 3's code used to execute it:  
I essentially copied what was made previously with the Alarm.png.  
I just replaced the values as needed.

With the tasks themselves it was easy to just place in the values and hit run.

```

bn = hh.BayesNet(
    ("A", "B"),
    ("B", "F"),
    ("C", "D"),
    ("D", "F"),
    seed=42,
) # Correct connections between nodes, just need to add probabilities like above

# Essentially follow the same format as above, just adjust the values to fit the practice.png
bn.P["A"] = pd.Series([False: .9, True: .1])

bn.P["B"] = pd.Series([
    (True, True): .4, # P(B = True | A=True) = 0.4
    (True, False): .6, # P(B = False | A=True) = 0.6
    (False, True): .9, # P(B = True | A=False) = 0.9
    (False, False): .1 # P(B = False | A=False) = 0.1
])

bn.P["C"] = pd.Series([False: .9, True: .1])

bn.P["D"] = pd.Series([
    (True, True): .3, # P(D = True | C=True) = 0.3
    (True, False): .7, # P(D = False | C=True) = 0.7
    (False, True): .6, # P(D = True | C=False) = 0.6
    (False, False): .4 # P(D = False | C=False) = 0.4
])

bn.P["F"] = pd.Series([
    (True, True, True): .7, # P(F=True | B=True, D=True) = 0.7
    (True, True, False): .3, # P(F=False | B=True, D=True) = 0.3

    (True, False, True): .4, # P(F=True | B=False, D=False) = 0.4
    (True, False, False): .6, # P(F=False | B=False, D=False) = 0.6

    (False, True, True): .2, # P(F=True | B=False, D=True) = 0.2
    (False, True, False): .8, # P(F=False | B=False, D=True) = 0.8

    (False, False, True): 0, # P(F=True | B=False, D=False) = 0
    (False, False, False): 1, # P(F=False | B=False, D=False) = 1
])

bn.prepare()

```

**Task 2:** What is the likelihood of F is true if A&D is true, B&C is False?

```

In [27]: bn.query("F", event={"A": True, "B": True, "C": False, "D": True})

Out[27]:
      P(F)
      F
False 0.3
True  0.7

```

**Task 3:** What is the likelihood of F is False if A&B is False, C&D is True?

```

In [28]: bn.query("F", event={"A": False, "B": False, "C": True, "D": True})

Out[28]:
      P(F)
      F
False 0.8
True  0.2

```