Week 12

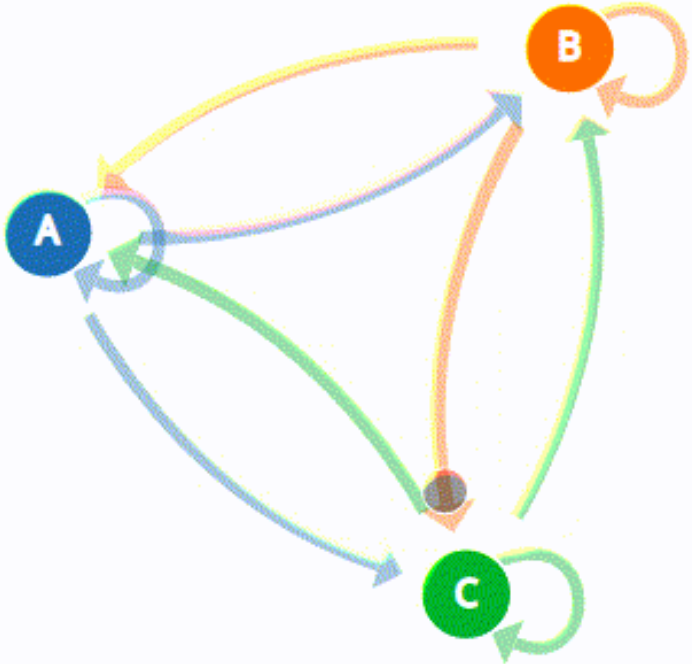
SIMULATION

Simulation

- Markov chain

Markov Chain

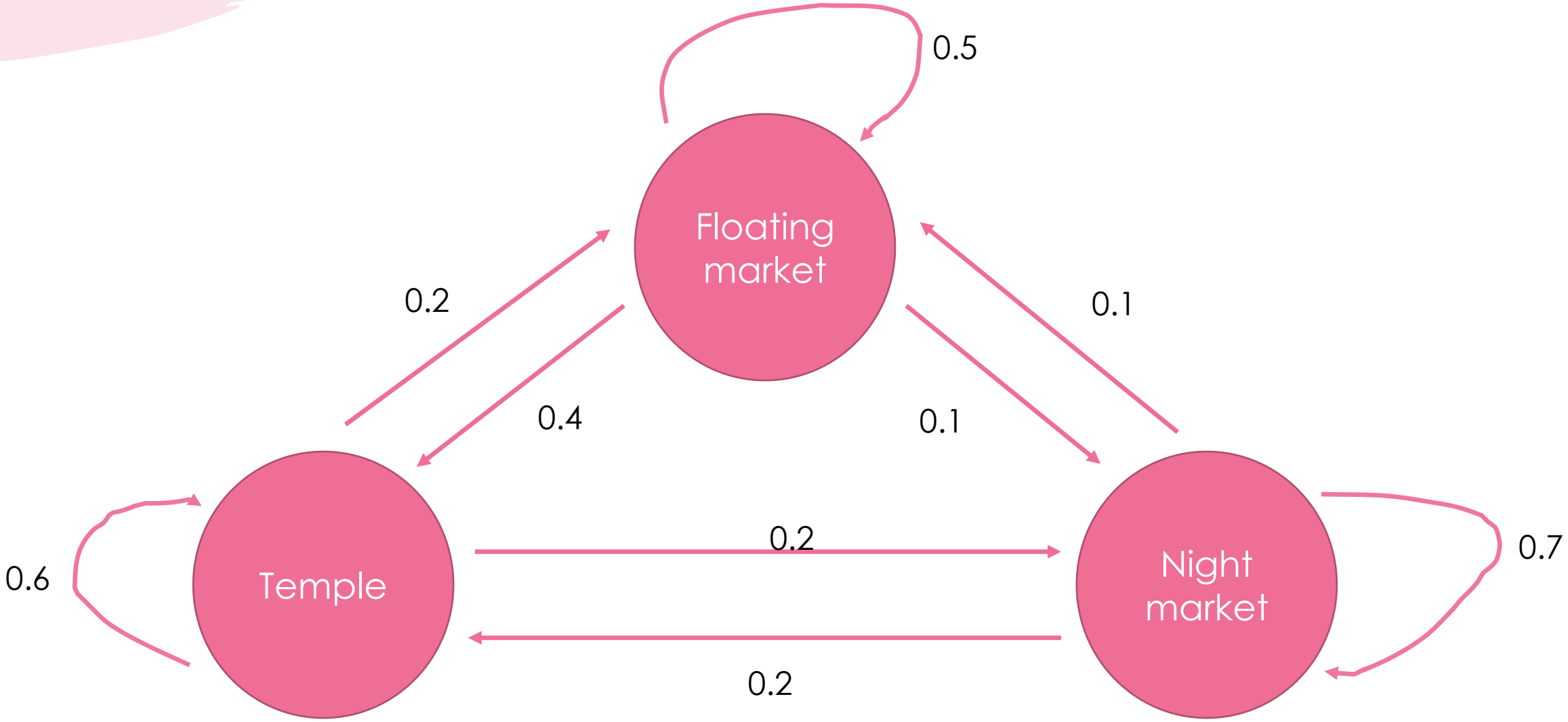
- It is a mathematical system that transition from one state to another according to certain probabilistic rules
- It is represented using a probabilistic automation and the changes of state of the system are called transitions
- The same information is represented by the transition matrix from time **n** to time **n+1**



Markov Chain Example

Travel case: when foreigners travel to Bangkok, they either go to temple, floating market and night market. According to historical data, if they go to temple, the next day it is 60% likely they will visit temple again, 20% visit to floating market and 20% visit to night market. On the other hand, if they travel to floating market, it is 50% chance they will remain visit at floating market, 40% go to temple and only 10% chance they will go to night market. Finally, when they travel to night market, there is 70% chance they will continue to visit night market, 20% chance visit temple and only 10% to floating market. Use Markov Chain simulation to find the probability on the next days.

Markov Chain – State diagram



Markov Chain – Transition Matrix

Next State

Current State

	Temple	Floating market	Night market
Temple	0.6	0.2	0.2
Floating market	0.4	0.5	0.1
Night market	0.2	0.1	0.7

Markov Chain in Python

```
import numpy as np
states = [“Temple”,”Floating”,”Night”]
transitionName = [[“TT”,”TF”,”TN”],[“FT”,”FF”,’FN”],[“NT”,”NF”,”NN”]
transitionMatrix = [[0.6, 0.2, 0.2] , [0.4, 0.5, 0.1] , [0.2, 0.1, 0.7]]

if sum(transitionMatrix[0])+sum(transitionMatrix[1])+sum(transitionMatrix[2]) != 3:
        print(“Somewhere, something went wrong. Transition matrix, perhap?”)
else:
      print(“All is gonna be okay, you should move on !!”)
```

Markov Chain in Python

```
def activity_forecast(days):
    activity_today = “Temple”
    print(“Start state: “ + activity_today)
    activityList = [activity_today]
    i = 1
    prob = 1
    while i != days:
        if activity_today == “Temple”:
            change = np.random.choice(transitionName[0], replace = True, p = transitionMatrix[0])
        if change == “TT”:
            prob = prob * 0.6
            activityList.append(“Temple”)
            pass
    elif change == “TF”
        prob = prob * 0.2
        activity_today = “Floating”
        activityList.append(“Floating”)
    else:
        prob = prob *0.2
        activity_today = “Night”
        activityList.append(“Night”)
```

Markov Chain in Python - Floating

```
    if activity_today == “Floating”:
        change = np.random.choice(transitionName[1], replace = True, p = transitionMatrix[1])
    if change == “FF”:
        prob = prob * 0.5
        activityList.append(“Floating”)
        pass
elif change == “FT”
    prob = prob * 0.4
    activity_today = “Temple”
    activityList.append(“Temple”)
else:
    prob = prob *0.1
    activity_today = “Night”
    activityList.append(“Night”)
```

Markov Chain in Python - Night

```
        if activity_today == “Night”:
            change = np.random.choice(transitionName[2], replace = True, p = transitionMatrix[2])
        if change == “NN”:
            prob = prob * 0.7
            activityList.append(“Night”)
            pass
    elif change == “NT”
        prob = prob * 0.2
        activity_today = “Temple”
        activityList.append(“Temple”)
    else:
        prob = prob *0.1
        activity_today = “Floating”
        activityList.append(“Floating”)
    i+=1
    print(“Possible state: “ + str(activityList))
    print(“End state after: “ + str(days) + “ days” + activity_today)
    print(“Probability of the possible sequence of states: “ + str(prob))

activity_forecast(2)
```

Markov Chain practice

Weather case: The Meteorological department would like to forecast the weather (sunny, rainy, cloudy) on the next day by using the historical data. Refer to historical data, it found that if today is sunny, they it has 70% chance it will remain sunny on tomorrow, 10% chance on rainy and 20% chance on cloudy. Moving to the next scenario, if today is rainy, it has only 5% chance on sunny on the next day, 35% chance cloudy and 60% chance on rainy. Last scenario, it has 50% chance of cloudy on next day, when today is cloudy, 30% chance of rainy and 20% chance on sunny.

a) Create the state diagram and transition matrix

b) Use Markov Chain simulation to find the probability of the next 2 days