Student Name: Sara Azarnoush
Student ID: 98170668

Computer Simulation

Assignment 1

---

1. Question 1

   (a) continuous - both (A static jitter buffer is hardware-based and is configured by the manufacturer. A dynamic jitter buffer is software-based and can be configured by the network administrator to adapt to changes in the network's delay.) - stochastic in nature

   (b) discrete - dynamic- stochastic

   (c) continuous - dynamic - stochastic

   (d) discrete - static - deterministic

2. Question 2

   The memoryless property (also called the forgetfulness property) means that a given probability distribution is independent of its history. Any time may be marked down as time zero.
   If a probability distribution has the memoryless property the likelihood of something happening in the future has no relation to whether or not it has happened in the past. The history of the function is irrelevant to the future.

   - Tossing a fair coin is an example of probability distribution that is memoryless. Every time you toss the coin, you have a 50 percent chance of it coming up heads. It doesn't matter whether or not the last five times you threw the dice it came up consistently tails; the probability of heads in the next throw is always going to be zero.
   - For a real life example, consider independent failures of computer hardware. When figuring the probability of a (new, independent) hardware failure it doesn't matter how frequently or when your hardware failed in the past. The probability it will fail five minutes from now is independent of the fact that it hasn't failed for three months. The probability distribution can be modeled by the exponential distribution or Weibull distribution, and it's memoryless.

   yes if the brocken parts are replaced and not old. time between car broke down in the future has no relation to whether or not it has happened broken down the past. The history of the function is irrelevant to the future. otherwise no if some parts are old and broken it will brocken again

3. Question 3

   (a) advantages

      i. Simulation allows to explore 'what if' questions and scenarios without having to experiment on the system itself.
      ii. It helps to identify bottlenecks in material, information and product flows.
      iii. It helps to gain insight into which variables are most important to system performance.

   disadvantages

      i. No matter how accurate the learning simulation is, there is always some scope for error and doubt when it comes to the re-creation of real life scenarios.

ii. Every technology gets outdated real quick. The cost involved in updating the technology is never cheap and in order to keep them as relevant as possible, simulators require regular updates and maintenance based on the changing trends of what is being taught.

iii. The quality of the analysis depends on the quality of the model and the skills of the modeller, who requires specialised training.

i. shortest job first serve: under 10 product shping queues in real world or computer systems. adv: no need to spend lots of time and more customer serves in short time. disadv: the one with long service time must wait alot

ii. first come first serve: shoping queues in real world. adv: justice in time - disadv: short work must wait long

iii. earliest deadline first:real life job and homeworks.adv: jobs done on time - disadv: not important jobs done first and may be no specific time for important one

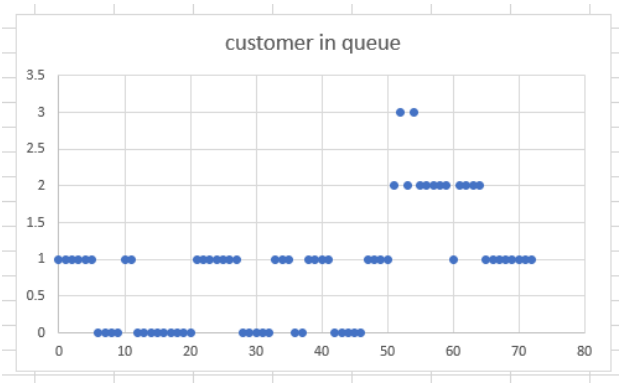
(b) i. Weather:
Weather forecasting uses simulations based on past data to predict extreme weather conditions such as hurricanes or cyclones.

ii. Sport:
Statistics are widely used as part of sport simulation to predict the outcome of events and the performance of individual sportspeople. Sports simulation can also be used to predict the outcome of games and events as well as for fantasy sports leagues. Biomechanics models can also be used to assist training, assess fatigue levels and their effect on performance and more.

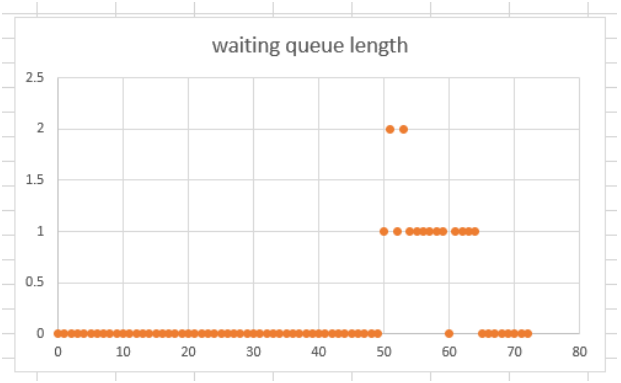
4. Question 4

	A	B	C	D	E	F	G
1	customer	time since last arrival	arrival time	serive time	time service begins	time wait	time service ends
2	1	0	0	6	0	0	6
3	2	10	10	2	10	0	12
4	3	11	21	7	21	0	28
5	4	12	33	3	33	0	36
6	5	5	38	4	38	0	42
7	6	9	47	5	47	0	52
8	7	3	50	5	60	10	65
9	8	1	51	2	52	1	54
10	9	2	53	6	54	1	60
11	10	8	61	7	65	4	72

(a) arrival time = time since last arrival + time since last arrival(i-1)
time service begins = max{time service ends(last one done), arrival time} (lifo=>if customer came later and has less ((time service ends(last one done in system) - time arrival)>=0) then it start first)
time service ends = time service begins + service time



(b) time service begins - arrival time = time wait in queue
queue length = number of customer waiting in queue in each time



(c) time service begins - time service ends(last one done) = idle time
(10-6)+(21-12)+(33-28)+(38-36)+(47-42)=25 (after 47 there is no free time)
25/72 = 0.3472 => 35%
35% of time its free and waste energy and time so its not optimal

(d) its not optimal. we can use other algorithm and check service time and time since last arrival to see which must come first not just last one come to system

5. Question 5

excel file with numbers and formulas next to pdf 5.xlsx

```python
import random
numbers = range(0,100)
time_last_arrival = []
service_time = []

for i in range(9): #first = 0
    rand = random.choice(numbers)
    if rand >=1 and rand <=3:
        time_last_arrival.append(random.choice([1,2,3,4]))
    elif rand >=4 and rand <=65: #3 + 62
        time_last_arrival.append(random.choice([5,6,7,8]))
    elif rand >=66 and rand <= 89: #65 +24
        time_last_arrival.append(random.choice([9,10,11]))
    else:
        x.append(rand)
        time_last_arrival.append(random.choice([12,13,14,15]))

for i in range(10):
    rand = random.choice(numbers)
    if rand >=1 and rand <=36:
        service_time.append(random.choice([3,4,5,6]))
    elif rand >=37 and rand <=79: #36 + 43
        service_time.append(random.choice([7,8,9,10]))
    else:
        service_time.append(random.choice([10,11,12]))

print("time_last_arrival: ", time_last_arrival)
print("service_time: ", service_time)

```

(a) wait time = time service begins - arrival time

(b) time service begins = max{time service ends(i-1), arrival time}

(c) time service ends = time service begins + service time

(d) time spends in system = time service ends - arrival time

(e) 1 customer -> time service begins - time service ends(i-1) = idle time
all -> sum of 10 customer

6. Question 6 (practical) there is a file next to pdf