CMR
INSTITUTE OF
TECHNOLOGY

Improvement Test – May. 2017-SCHEME OF SOLUTION

Sub:	System Modeling and Simulation							Code:	10CS82
Date:	25/ 05/2017	Duration:	90 mins	Max Marks:	50	Sem:	VIII	Branch:	ISE/CSE

Note: Answer any five questions:

1. Explain the replication method for steady – state simulations. **10M**

If initialization bias in the point estimator has been reduced to a negligible level (through some combination of intelligent initialization and deletion), then the method of independent replications can be used to estimate point-estimator variability and to construct a confidence interval. The basic idea is simple: Make R replications, initializing and deleting from each one the same way.

If, however, significant bias remains in the point estimator and a large number of replications are used to reduce point-estimator variability, the resulting confidence interval can be misleading. This happens because *bias is not affected by the number of replications* (R); it is affected only by deleting more data (i.e., increasing T_0) or extending the length of each run (i.e., increasing T_E). Thus, increasing the number of replications (R) could produce shorter confidence intervals around the "wrong point." Therefore, it is important to do a thorough job of investigating the initial-condition bias.

If the simulation analyst decides to delete d observations of the total of n observations in a replication, then the point estimator of θ is $\bar{Y}..(n, d)$, defined by Equation (11.25)—that is, the point estimator is the average of the remaining data. The basic raw output data, $\{Y_{rj}, r = 1,\ldots, R; j = 1,\ldots, n\}$, are exhibited in Table 11.7. Each Y_{rj} is derived in one of the following ways:

Case 1

Y_{rj} is an individual observation from within replication r; for example, Y_{rj} could be the delay of customer j in a queue, or the response time to job j in a job shop.

Case 2

Y_{rj} is a batch mean from within replication r of some number of discrete-time observations.

Case 3

Y_{rj} is a batch mean of a continuous-time process over time interval j;

In Case 1, the number d of deleted observations and the total number of observations n might vary from one replication to the next, in which case replace d by d_r and n by n_r. For simplicity, assume that d and n are constant over replications. In Cases 2 and 3, d and n will be constant.

Table 11.7 Raw Output Data from a Steady-State Simulation

Replication	Observations 1	...	d	$d+1$	...	n	Replication Averages
1	$Y_{1,1}$	...	$Y_{1,d}$	$Y_{1,d}+1$	...	$Y_{1,n}$	$\bar{Y}_{1}.(n, d)$
2	$Y_{2,1}$	...	$Y_{2,d}$	$Y_{2,d}+1$	...	$Y_{2,n}$	$\bar{Y}_{2}.(n, d)$
.	.		.	.		.	.
.	.		.	.		.	.
.	.		.	.		.	.
R	Y_{R1}	...	Y_{Rd}	$Y_{R,d+1}$	...	Y_{Rn}	$\bar{Y}_{R}.(n, d)$
	$Y_{\cdot 1}$	...	$\bar{Y}_{\cdot d}$	$\bar{Y}_{\cdot ,d+1}$	...	$\bar{Y}_{\cdot n}$	$\bar{Y}..(n, d)$

When using the replication method, each replication is regarded as a single sample for the purpose of estimating θ. For replication r, define

$$\bar{Y}_r.(n,d) = \frac{1}{n-d}\sum_{j=d+1}^{n} Y_{rj} \tag{11.33}$$

as the sample mean of all (nondeleted) observations in replication r. Because all replications use different random-number streams and all are initialized at time 0 by the same set of initial conditions (I_0), the replication averages

$$\bar{Y}_1.(n,d), \ldots, \bar{Y}_R.(n,d)$$

are independent and identically distributed random variables—that is, they constitute a random sample from some underlying population having unknown mean

$$\theta_{n,d} = E\,[\bar{Y}_r.(n,d)] \tag{11.34}$$

The overall point estimator, given in Equation (11.25), is also given by

$$\bar{Y}..(n,d) = \frac{1}{R}\sum_{r=1}^{R} \bar{Y}_r.(n,d) \tag{11.35}$$

as can be seen from Table 11.7 or from using Equation (11.24). Thus, it follows that

$$E[\bar{Y}..(n,d)] = \theta_{n,d}$$

also. If d and n are chosen sufficiently large, then $\theta_{n,d} \approx \theta$, and $\bar{Y}..(n,d)$ is an approximately unbiased estimator of θ. The bias in $\bar{Y}..(n,d)$ is $\theta_{n}, d-\theta$.

For convenience, when the value of n and d are understood, abbreviate $\bar{Y}_r.(n,d)$ (the mean of the undeleted observations from the rth replication) and $\bar{Y}..(n,d)$ (the mean of $\bar{Y}_1.(n,d), \ldots, \bar{Y}_R.(n,d)$ by $\bar{Y}_r.$ and $\bar{Y}..$, respectively. To estimate the standard error of $\bar{Y}..$, first compute the sample variance,

$$S^2 = \frac{1}{R-1}\sum_{r=1}^{R}(\bar{Y}_r. - \bar{Y}..)^2 = \frac{1}{R-1}\left(\sum_{r=1}^{R}\bar{Y}_r.^2 - R\bar{Y}..^2\right) \tag{11.36}$$

The standard error of $\bar{Y}..$ is given by

$$\text{s.e.}(\bar{Y}..) = \frac{S}{\sqrt{R}} \tag{11.37}$$

A $100(1-\alpha)\%$ confidence interval for θ, based on the t distribution, is given by

$$\bar{Y}.. - t_{\alpha/2,R-1}\frac{S}{\sqrt{R}} \le \theta \le \bar{Y}.. + t_{\alpha/2,R-1}\frac{S}{\sqrt{R}} \tag{11.38}$$

	where $t_{\alpha/2,R-1}$ is the $100(1-\alpha/2)$ percentage point of a t distribution with $R-1$ degrees of freedom. This confidence interval is valid only if the bias of $\bar{Y}_{..}$ is approximately zero. As a rough rule, the length of each replication, beyond the deletion point, should be at least ten times the amount of data deleted. In other words, $(n-d)$ should at least $10d$ (or more generally, T_E should be at least $10T_0$). Given this run length, the number of replications should be as many as time permits, up to about 25 replications. Kelton [1986] established that there is little value in dividing the available time into more than 25 replications, so, if time permits making more than 25 replications of length $T_0 + 10T_0$, then make 25 replications of longer than $T_0 + 10T_0$, instead. Again, these are rough rules that need not be followed slavishly.	
2	a) What are world views? Explain the types and explain three phase approach in detail. When using a simulation_ package or even when doing a manual simulation, a modeler adopts a world view or orientation for developing a model. The most prevalent world views are the event-scheduling world view, as discussed m the previous section, the process interaction world view, and the activity-scanning world view. Even if a particular package does not directly support one or more of the world views, understanding the different approaches could suggest alternative ways to model a given system. With the activity-scanning approach, a modeler concentrates on the activities of a model and those conditions, simple or complex, that allow an activity to begin. At each clock advance, the conditions for each activity are checked, and, if the conditions are true, then the corresponding activity begins. Proponents claim that the activity-scanning approach is simple in concept and leads to modular models that are more easily maintained, understood, and modified by other analysts at later times. They admit, however, that the repeated scanning to discover whether an activity can begin results in slow runtime on computers. Thus, the pure activity-scanning approach has been modified (and made conceptually somewhat more complex) by what is called the three-phase approach, which combines some of the features of event scheduling with activity scanning to allow for variable time advance and the avoidance of scanning when it is not necessary, but keeps the main advantages of the activity-scanning approach. In the three-phase approach, events are considered to be activities of duration zero time units. With this definition, activities are divided into two categories, which are called B and C. B activities activities bound to occur; all primary events and unconditional activities. C activities activities or events that is conditional upon certain conditions-being true. The B-type activities and events can be scheduled ahead of time, just as in the event-scheduling approach. This allows variable time advance. The FEL contains only B-type events. Scanning to learn whether any C-type activities can begin or C-type events occur happens only at the end of each time advance, after all B-type events have completed. In summary, with the three-phase approach, the simulation proceeds with repeated execution of the 3 phases until it is completed; Phase A Remove the imminent event from the FEL and advance the clock to its event time. Remove from the FEL any other events that have the same event time. Phase B Execute all B-type events that were removed from the FEL. (This could free a number of resources or otherwise change system state.) Phase C Scan the conditions that trigger each C-type activity and activate any whose conditions are sent. Rescan until no additional C-type activities can begin and no events occur. The three-phase approach improves the execution efficiency of the activity-scanning method. In addition, proponents claim that the activity scanning and three-phase approaches are particularly good at handling complex resource problems in which various combinations of resources are needed to	5M

accomplish different tasks. These approaches guarantee that all resources being freed at a given simulated time will all be freed before any available resources are reallocated to new tasks.

b) What is a model? Explain different types of models with examples.

A model is defined as a representation of a system for the purpose of studying the system.

TYPES OF MO DELS

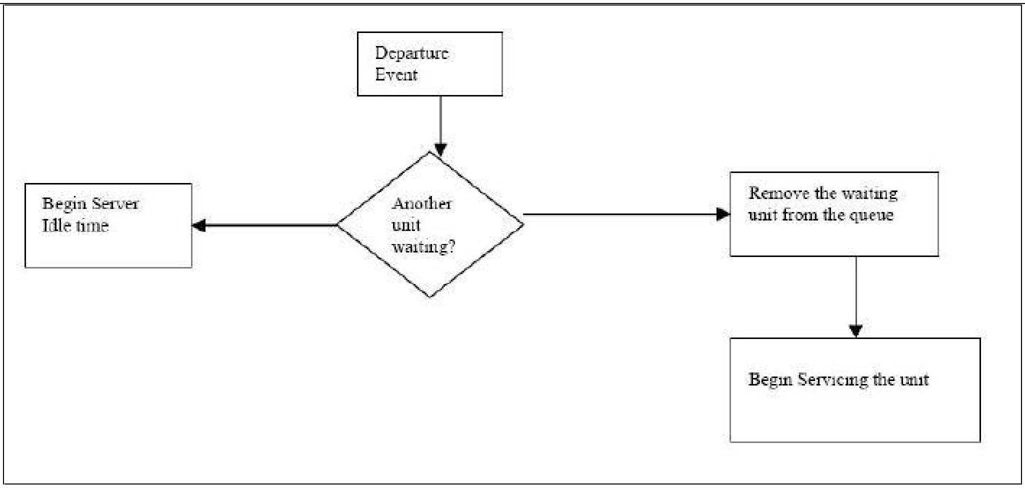
Models can be classified as being mathematical or physical. A mathematical model uses symbolic notation and mathematical equations to represent a system. A simulation model is a particular type of mathematical model of a system. Simulation models may be further classified as being static or dynamic, deterministic or stochastic, and discrete or continuous. A static simulation model, sometinles called a Monte Carlo simulation, represents a, system at a particular point in time. Dynamic simulation models represent systems as they change over time.The simulation of a bank from 9:00 A.M. to 4:00 P.M. is an example of a dynamic simulation. 5M

Simulation models that contain no random variables are classified as deterministic. Deterministic models have a known set of inputs, which will result in a unique set of outputs. Deterministic arrivals would occur at a dentist's office if all patients arrived at the scheduled appointment time. A stochastic simulation model has one or more random variables as inputs. Random inputs lead to random outputs. Since the outputs are random, they can be considered only as estimates of the true characteristics of a model. The simulation of a bank would usually involve random inter arrival times and random service times. Thus, in a stochastic simulation, the output measures-the average number of people waiting, the average waiting time of a customer-must be treated as statistical estimates of the true characteristics of the system.

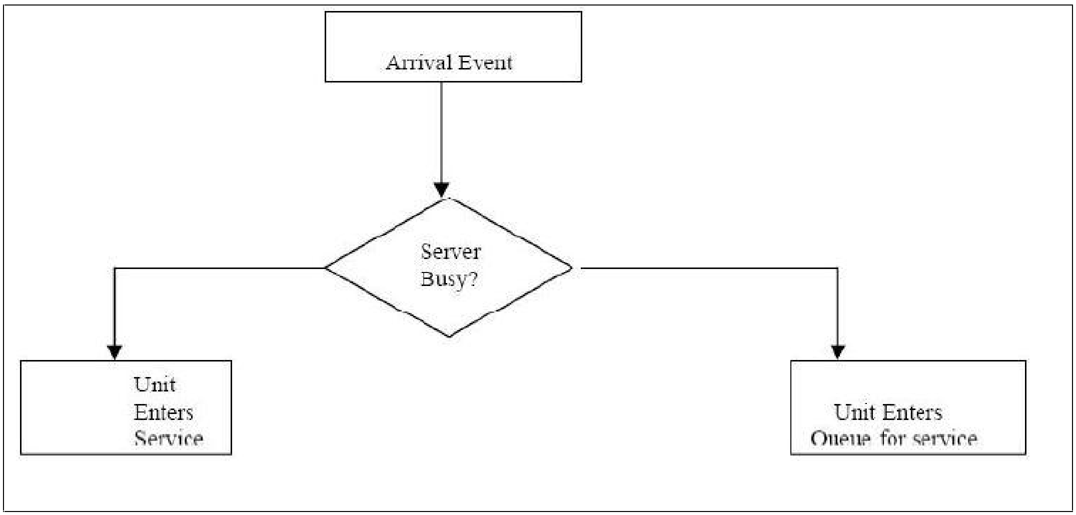
Discrete and continuous models are defined in an analogous manner. However, a discrete simulation model is not always used to model a discrete system, nor is a continuous simulation model always used to model a continuous system. Tanks and pipes are modeled discretely by some software vendors, even though we know that fluid flow is continuous. In addition, simulation models may be mixed, both discrete and continuous. The choice of whether to use a discrete or continuous {or both discrete and continuous) simulation model is a function of the characteristics of the system and the objective of the study. Thus, a communication channel could be modeled discretely if the characteristics and movement of each message were deemed important. Conversely, if the flow of messages in aggregate over the channel of importance, modeling the system via continuous simulation could be more appropriate. The models considered in this text are discrete, dynamic, and stochastic.

3 a) With the help of flowchart explain simulation of single channel queuing system. 5M

-Departure event flow diagram with explanation-2M



-Arrival event flow diagram with explanation-3M



***Either all the 4 diagrams can be drawn or Minimum 2 diagrams is needed.**

b) Explain Simulation in GPSS with a neat block diagram.

GPSS is a highly structured, special-purpose simulation programming language based on the process-interaction approach and oriented toward queuing systems. A block diagram provides a convenient way to describe the system being simulated. There are over 40 standard blocks in GPSS. Entities called transactions may be viewed as flowing through the block diagram. Blocks represent events, delays, and other actions that affect transaction flow. Thus, GPSS can be used to model any situation where transactions (entities, customers, units of traffic) are flowing through a system (e.g., a network of queues, with the queues preceding scarce resources). The block diagram is converted to block statements, control statements are added, and the result is a GPSS model. 5M

The first version of GPSS was released by IDM in 1961. It was the first process-interaction simulation language and became popular; it has been implemented a new and improved by many parties since 1961, with GPSS/H being the most widely used version in use today. GPSS/H is a product of Wolverine SoftWare Corporation, Annandale, VA (Banks, Carson, and Sy, 1995; Henriksen, 1999). It is a flexible, yet powerful tool for simulation. Unlike the original IDM implementation, GPSS/H includes built-in file and screen 1/0, use of an arithmetic expression as a block operand, an interactive

Debugger, faster execution, expanded control statements, ordinary variables and arrays, a floating point clock, built-in math functions, and built-in random-variate generators. The animator for GPSS/H is Proof Animation™, another product of Wolverine Software Corporation (Henriksen, 1999). Proof Animation provides a 2-D animation, usually based on a scale drawing. It can run in post processed mode (after the simulation has finished running) or concurrently. In post processed mode, the animation is driven by two files: the layout file for the static background, and a trace file that contains Commands to make objects move and produce other dynamic events. It can work with any simulation package that can write the ASCII trace file. Alternately, it can run concurrently with the simulation by sending (he trace file commands as messages, or it can be controlled directly by using its DLL (dynamic link library) version.

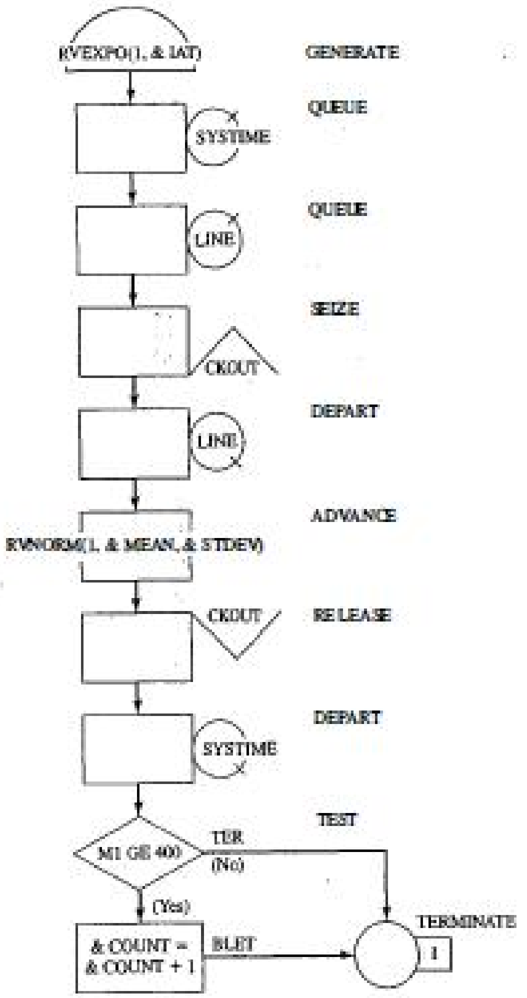


Figure 4.10 GPSS block diagram for the singleserver queue simulation.

4 Consider the following inter arrival times and service times. Using time advance prepare a simulation table based on the following activities and stop the simulation when clock reaches 25. **10M**

Inter arrival Time	4	5	2	8	3	7
Service Time	5	3	4	6	2	7

-For defining the system states like LQ(t),LS(t) – 2M
-For defining the Future Event List-4M
-For Updating the Cumulative statistics like B and MQ-2M
-For Simulation table defining all the above entities-2M

Clock	System State		Future Event List	Cumulative Statistics	
	LQ(t)	LS(t)		B	MQ
0	0	1	(A,C2,4)(D,C1,5)(E,25)	0	0
4	1	1	(D,C1,5)(A,C3,9)(E,25)	4	1
5	0	1	(D,C2,8)(A,C3,9)(E,25)	5	1
8	0	0	(A,C3,9)(E,25)	8	1
9	0	0	(A,C4,11)(D,C3,13)(E,25)	8	1
11	1	1	(D,C3,13) (A,C5,19)(E,25)	11	1
13	0	1	(D,C4,19)(A,C5,19),(E,25)	13	1
19	0	1	(D,C5,21)(A,C6,22)(E,25)	19	1
21	0	0	(A,C6,22)(E,25)	21	1
22	0	1	(D,C6,29)(A,C7,29)(E,25)	21	1
25	0	1	(D,C6,29)(A,C7,29)(E,25)	25	1

5. Let the arrival distribution be uniformly distributed between 1 to 8 min. Develop a simulation table for 6 customers. The service time distributions are as follows. 10M

Service Time	1	2	3	4	5	6
Probability	0.10	0.20	0.30	0.25	0.10	0.05

Consider the following random numbers for inter Arrival Times and Service Times.
Random No for IAT: 502, 617, 391, 159,999 and 752
Random No for Service Times: 24, 15, 62, 73, 42, 17.
Find the average waiting time and probability of idle time of server from the simulation table.

-For finding the cumulative probability and random numbers for IAT-2M

IAT	Probability	Cumulative Probability	Random No Assessment
1	0.125	0.125	0-125
2	0.125	0.250	126-250
3	0.125	0.375	251-375
4	0.125	0.500	376-500
5	0.125	0.625	501-625
6	0.125	0.750	626-750
7	0.125	0.875	751-875
8	0.125	1	876-999

-For finding the cumulative probability and random numbers for service times-2M

ST	Probability	Cumulative Probability	Random No Assessment
1	0.10	0.10	1-10
2	0.20	0.30	11-30
3	0.30	0.60	31-60
4	0.25	0.85	61-85
5	0.10	0.95	86-95
6	0.05	1	96-99

-Main Simulation table-6M
*The marks split up for this table is as shown below
-For finding the Inter-arrival times and arrival times-1M
- For finding the service times from the random numbers-1M
-For finding the time service begins and time service ends-2M each

Customer	IAT	AT	ST	Time Service Begins	Waiting Time	Time Service Ends	Time Customer Spend in system	Idle time of server
1	-	0	2	0	0	2	2	0
2	5	5	2	5	0	7	2	3
3	5	10	4	10	0	14	4	3
4	4	14	4	14	0	18	4	0
5	5	16	3	18	2	21	5	0
6	8	24	2	24	0	26	2	3
Total					2			9

-For finding the average waiting time and Probability of idle time of server-2M

Average WT = Total WT/Total No of customers =2/6 =0.33 Min
Probability of idle time of server = Total idle time/Total run time of simulation = 9/26 = 0.34Min

6 Develop a simulation table for Able-baker call center problem assuming that able can do the job better than baker. The time between arrivals and service times are as shown below. Simulate for 5 days. 10M

Service Time of Able	2	3	4	5
Probability	0.20	0.25	0.25	0.30
Service Time of Baker	3	4	5	6
Probability	0.10	0.30	0.25	0.35

Time b/n arrivals	1	2	3	4
Probability	0.35	0.20	0.30	0.15

Consider the following random numbers.
Random no for Arrivals: 42,74,80,34,68
Random no for Service Time: 89,13,50,49,88

-For Finding the cumulative probability and random numbers for inter-Arrival Times-1M

Inter-Arrival time	Probability	Cumulative Probability	Random No Assessment
1	0.35	0.35	0-35
2	0.20	0.55	36-65
3	0.30	0.85	56-85
4	0.15	1.00	86-99

-For Finding the Cumulative probability and random no for service times of Able and Baker-1M

ST of Able	Probability	Cumulative Probability	Random No Assessment	ST of Baker	Probability	Cumulative Probability	Random No Assessment
2	0.20	0.20	1-20	3	0.10	0.10	0-10
3	0.25	0.45	21-45	4	0.30	0.40	11-40
4	0.25	0.70	46-70	5	0.25	0.65	41-65
5	0.30	1.00	71-99	6	0.35	1.00	66-99

-Main Simulation Table-8M.
*The Marks Split up for this table is as shown below.
-For Finding the Inter-Arrival Times from the random numbers-1M
-For Finding the arrival Times-1M
-For Finding the Service Times from the random numbers-1M
-For Finding the available server and time service begins-2M
-For Service completion time and time in the system-3M

Caller No	IAT	AT	When Able Available	Whwn Baker Available	Server Choosen	ST	Time Service Begins	Service Completion time Able	Baker	Caller Delay	Time in the system
1	-	0	0	0	Able	5	0	5	0		5
2	2	2	5	0	Baker	4	2	5	6		4
3	3	5	9	6	Able	4	5	9	6		4
4	3	8	9	13	Baker	5	8	9	13		5
5	1	9	14	13	Able	5	9	14	13		5

7 Explain the characteristics of queuing system. Explain different queuing notations. 10M

characteristics of queuing system

Key elements of queuing systems:

Customer: refers to anything that arrives at a facility and requires service, e.g., people, machines, trucks, emails.

Server: refers to any resource that provides the requested service, e.g., repairpersons, retrieval machines, runways at airport.

Calling population: the population of potential customers, may be assumed to be finite or infinite.

- **Finite population model**: if arrival rate depends on the number of customers being served and waiting, e.g., model of one corporate jet, if it is being repaired, the repair arrival rate becomes zero.
- **Infinite population model**: if arrival rate is not affected by the number of

customers being served and waiting, e.g., systems with large population of potential customers.

 System Capacity: a limit on the number of customers that may be in the waiting line or system.

 Limited capacity, e.g., an automatic car wash only has room for *10* cars to wait in line to enter the mechanism.

 Unlimited capacity, e.g., concert ticket sales with no limit on the number of people allowed to wait to purchase tickets.

 For infinite-population models:

 In terms of interarrival times of successive customers.

 Random arrivals: interarrival times usually characterized by a probability distribution.

 Most important model: Poisson arrival process (with rate *l*), where A_n represents the interarrival time between customer *n-1* and customer *n*, and is exponentially distributed (with mean *1/l*).

 Scheduled arrivals: interarrival times can be constant or constant plus or minus a small random amount to represent early or late arrivals.

 e.g., patients to a physician or scheduled airline flight arrivals to an airport.

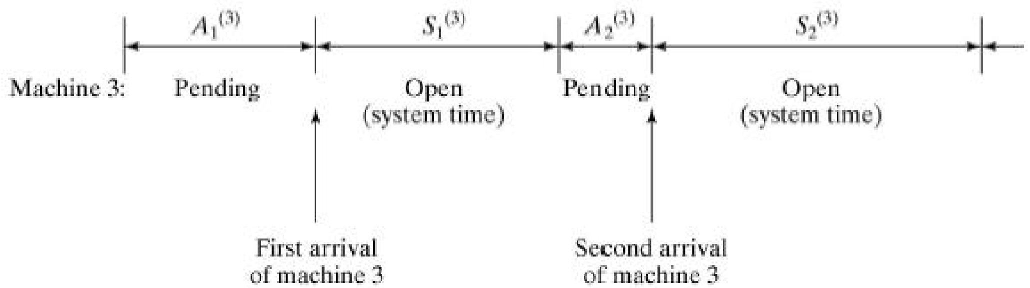
 At least one customer is assumed to always be present, so the server is never idle, e.g., sufficient raw material for a machine.

 For finite-population models:

 Customer is pending when the customer is outside the queueing system, e.g., machine-repair problem: a machine is "pending" when it is operating, it becomes "not pending" the instant it demands service form the repairman.

 Runtime of a customer is the length of time from departure from the queueing system until that customer's next arrival to the queue, e.g., machine-repair problem, machines are customers and a runtime is time to failure.

 Let $A_1^{(i)}$, $A_2^{(i)}$, … be the successive runtimes of customer i, and $S_1^{(i)}$, $S_2^{(i)}$ be the corresponding successive system times:



 Queue behavior: the actions of customers while in a queue waiting for service to begin, for example:

 Balk: leave when they see that the line is too long.

- Renege: leave after being in the line when its moving too slowly.
- Jockey: move from one line to a shorter line.

Queue discipline: the logical ordering of customers in a queue that determines which customer is chosen for service when a server becomes free, for example:

- First-in-first-out (FIFO)
- Last-in-first-out (LIFO)
- Service in random order (SIRO)
- Shortest processing time first (SPT)
- Service according to priority (PR).

Service times of successive arrivals are denoted by S_1, S_2, S_3.

- May be constant or random.
- $\{S_1, S_2, S_3, \ldots\}$ is usually characterized as a sequence of independent and identically distributed random variables, e.g., exponential, Weibull, gamma, lognormal, and truncated normal distribution.
- Each service center consists of some number of servers, c, working in parallel, upon getting to the head of the line, a customer takes the 1^{st} available server.

8 Consider the loading times, weighing times and travelling times as follows. 10M

Loading Time	10	5	10	10	5	10	5
Weighing Time	12	16	12	12	16	12	12
Travelling Time	40	60	40	80	100	40	

Assume that 2 trucks are at loading ,1 on scale and remaining 3 are at loader queue at time 0 and each truck is loaded by one of the two loaders. Using time advance algorithm simulate by considering stopping time after 10 iterations. Calculate the total busy time of loaders, scale, average loader and scale utilization.

Clock	System States				List		FEL	Cumulative Statistics	
	LQ(t)	L(t)	WQ(t)	W(T)	LQ(t)	WQ(t)		BL	BS
0	3	2	0	1	DT4,DT5 ,DT6	0	(EL,DT3,5)(EL,DT2,10) (EW,DT1,12)	0	0
5	2	2	1	1	DT5,DT6	DT3	(EL,DT2,10) (EW,DT1,12) (EL,DT4,15)	10	5
10	1	2	2	1	DT6	DT3,DT2	(EW,DT1,12) (EL,DT4,15) (EL,DT5,20)	20	10
12	1	2	1	1	DT6	DT2	(EL,DT4,15)(EL,DT5,20) (EW,DT3,28)(ALQ,DT1,52)	24	12
15	0	2	2	1	-	DT2,DT4	(EL,DT5,20)(EL,DT6,20) (EW,DT3,28)(ALQ,DT1,52)	30	15
20	0	0	4	1	-	DT2,DT4, DT5,DT6	(EW,DT3,28)(ALQ,DT1,52)	40	20
28	0	0	3	1	1	DT4,DT5 DT6	(EW,DT2,40)(ALQ,DT1,52) (ALQ,DT3,88)	40	28
40	0	0	2	1	-	DT5,DT6	(EW,DT4,52)(ALQ,DT1,52) (ALQ,DT2,80)(ALQ,DT3,88)	40	40
52	0	1	1	1	-	DT6	(EL,DT1,62)(EW,DT5,68) (ALQ,DT2,80)(ALQ,DT3,88) (ALQ,DT4,132)	40	52
62	0	0	2	1	-	DT6,DT1	(EW,DT5,68)(ALQ,DT2,80) (ALQ,DT3,88)(ALQ,DT4,132)	50	62