ANSWER KEY for IA- II

Subject:System Simulation & Modeling **Subject Code:13MCA52**

1) System state [LQ(t), L(t), WQ(t), W(t)]
   - LQ(t) = number of trucks in loader queue
   - L(t) = number of trucks (0, 1, or 2) being loaded
   - WQ(t) = number of trucks in weigh queue
   - W(t) = number of trucks (0 or 1) being weighed, all at simulation time t

   Event notices :
   - (ALQ, t, DTi), dump truck i arrives at loader queue (ALQ) at time t
   - (EL, t, DTi), dump truck i ends loading (EL) at time t
   - (EW, t, DTi), dump truck i ends weighing (EW) at time t

   Entities : The six dump trucks (DT 1, … , DT 6)

Simulation Table for Dump Truck Operation

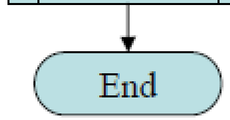
Clock	System State				Lists			Cumulative Statistics	
t	$LQ(t)$	$L(t)$	$WQ(t)$	$W(t)$	Loader Queue	Weigh Queue	Future Event List	B_L	B_S
0	3	2	0	1	DT4 DT5 DT6		(EL, 5, DT3) (EL, 10, DT2) (EW, 12, DT1)	0	0
5	2	2	1	1	DT5 DT6	DT3	(EL, 10, DT2) (EL, 5 + 5, DT4) (EW, 12, DT1)	10	5
10	1	2	2	1	DT6	DT3 DT2	(EL, 10, DT4) (EW, 12, DT1) (EL, 10 + 10, DT5)	20	10
10	0	2	3	1		DT3 DT2 DT4	(EW, 12, DT1) (EL, 20, DT5) (EL, 10 + 15, DT6)	20	10
12	0	2	2	1		DT2 DT4	(EL, 20, DT5) (EW, 12 + 12, DT3) (EL, 25, DT6) (ALQ, 12 + 60, DT1)	24	12
20	0	1	3	1		DT2 DT4 DT5	(EW, 24, DT3) (EL, 25, DT6) (ALQ, 72, DT1)	40	20
24	0	1	2	1		DT4 DT5	(EL, 25, DT6) (EW, 24 + 12, DT2) (ALQ, 72, DT1) (ALQ, 24 + 100, DT3)	44	24
25	0	0	3	1		DT4 DT5 DT6	(EW, 36, DT2) (ALQ, 72, DT1) (ALQ, 124, DT3)	45	25
36	0	0	2	1		DT5 DT6	(EW, 36 + 16, DT4) (ALQ, 72, DT1) (ALQ, 36 + 40, DT2) (ALQ, 124, DT3)	45	36

6)

52	0	0	1	1		$DT6$	$(EW, 52+12, DT5)$ $(ALQ, 72, DT1)$ $(ALQ, 76, DT2)$ $(ALQ, 52+40, DT4)$ $(ALQ, 124, DT3)$	45	52

Continued

Clock	*System State*				*Lists*			*Cumulative Statistics*	
t	$LQ(t)$	$L(t)$	$WQ(t)$	$W(t)$	*Loader Queue*	*Weigh Queue*	*Future Event List*	B_L	B_S
64	0	0	0	1			$(ALQ, 72, DT1)$ $(ALQ, 76, DT2)$ $(EW, 64+16, DT6)$ $(ALQ, 92, DT4)$ $(ALQ, 124, DT3)$ $(ALQ, 64+80, DT5)$	45	64
72	0	1	0	1			$(ALQ, 76, DT2)$ $(EW, 80, DT6)$ $(EL, 72+10, DT1)$ $(ALQ, 92, DT4)$ $(ALQ, 124, DT3)$ $(ALQ, 144, DT5)$	45	72
76	0	2	0	1			$(EW, 80, DT6)$ $(EL, 82, DT1)$ $(EL, 76+10, DT2)$ $(ALQ, 92, DT4)$ $(ALQ, 124, DT3)$ $(ALQ, 144, DT5)$	49	76



Time itself is represented by a variable called CLOCK
The definition of the model components provides a static description of the model. In addition, a description of the dynamic relationships and interactions between the components is also needed.

A discrete-event simulation (hereafter called a simulation) proceeds by producing a sequence of system snapshots (or system images) that represent the evolution of the system through time.

A given snapshot at a given time (CLOCK = t) includes not only the system state at time t**,** but also a list (the FEL) of all activities currently in progress and when each such activity will end, the status of all entities and current membership of all sets, plus the current values of cumulative statistics and counters that will be used to calculate summary statistics at the end of the simulation.

The mechanism for advancing simulation time and guaranteeing that all events occur in correct chronological order is based on the future event list (FEL).

This list contains all event notices for events that have been scheduled to occur at a future time.
Scheduling a future event means that, at the instant an activity begins, its duration is computed or drawn as a sample from a statistical distribution; and that the end-activity event, together with its event time, is placed on the future event list.
In the real world, most future events are not scheduled but merely happen—such as random breakdowns or random arrivals.
In the model, such random events are represented by the end of some activity, which in turn is represented by a statistical distribution.

Time *t* is the value of CLOCK, the current value of simulated time.

- The event associated with time t_1 is called the imminent event; that is, it is the next event that will occur.
- After the system snapshot at simulation time CLOCK = *t* has been updated, the CLOCK is advanced to simulation time CLOCK = t_1, the imminent event notice is removed from the FEL, and the event is executed.
- Execution of the imminent event means that a new system snapshot for time t_1 is created, one based on the old snapshot at time *t* and the nature of the imminent event.
- At time t_1, new future events may or might not be generated, but if any are, they are scheduled by creating event notices and putting them into their proper position on the FEL.
- After the new system snapshot for time t_1 has been updated, the clock is advanced to the time of the new imminent event and that event is executed.
- This process repeats until the simulation is over.

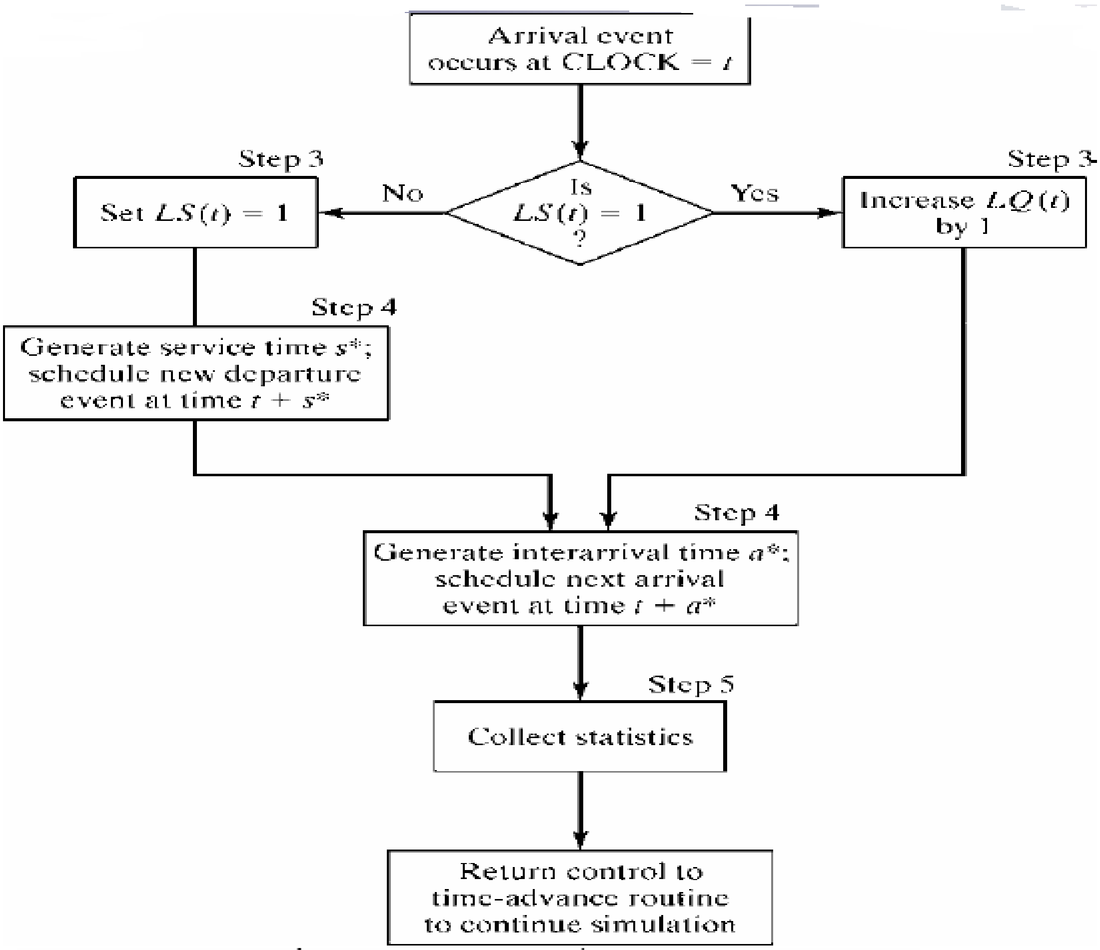
Event-scheduling / time-advance algorithm

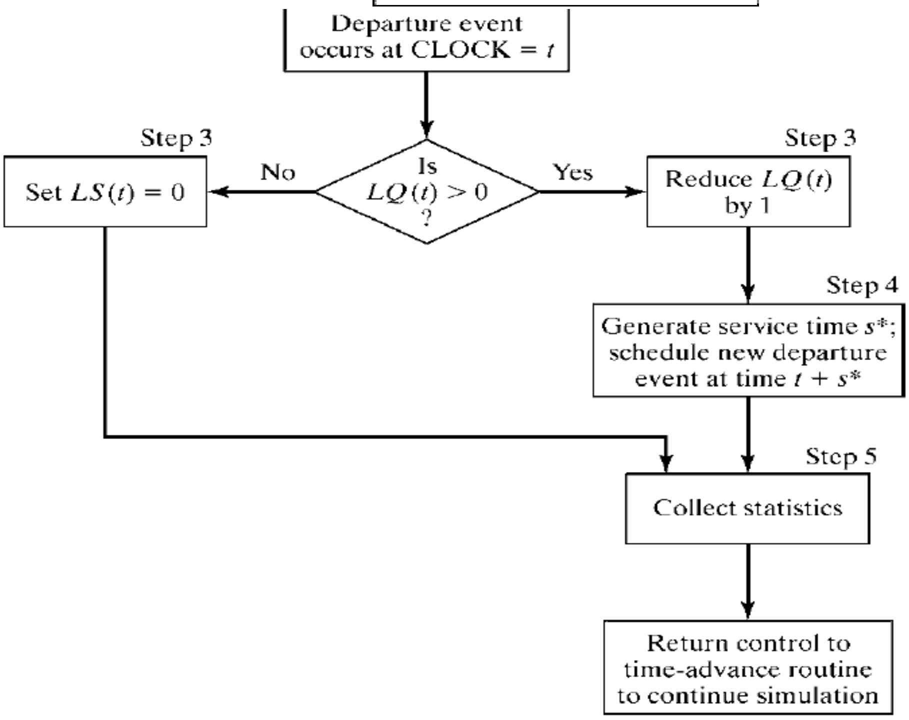
Step1 Remove the event notice for the imminent event (event 3, time t_1) from FEL
Step 2 Advance CLOCK to imminent event time (i.e. advance clock from t to t_1)
Step 3 Execute imminent event: update system state, change entity attributes and set membership as needed.
Step 4 Generate future events (if necessary) and place their event notices on FEL, ranked by event time. (Example: Event 4 to occur at time t*, where $t_2 < t^* < t_3$)
Step 5 Update cumulative statistics and counters

Every simulation must have a stopping event, here called *E*, which defines how long the simulation will run. There are generally two ways to stop a simulation:

1. At time 0, schedule a stop simulation event at a specified future time T_E. Thus, before simulating, it is known that the simulation will run over the time interval [0, T_E]. Example: Simulate a job shop for T_E = 40 hours.

2. Run length T_E is determined by the simulation itself. Generally, T_E is the time of occurrence of some specified event *E*. Examples: T_E is the time of the 100th service completion at a certain service center. T_E is the time of breakdown of a complex system. T_E is the time of disengagement or total kill (whichever occurs first) in a combat simulation. T_E is the time at which a distribution center ships the last carton in a day's orders.





3)a)**List Processing in Simulation:**

- Most simulations involve *lists*
  - Queues, event list, others
  - A list is composed of *records*
- *Record*: Usually corresponds to an object in the list
  - By convention, a record is represented as a row in a two-dimensional array (matrix) representing the list
  - A person in a queue list, an event in the event list
  - A record is composed of one or more *attributes*
- *Attribute*: A data field of each record
  - By convention, attributes are in columns
  - Examples of records (lines) of attributes (columns):
    - Queue list: [time of arrival, customer type, service requirement, priority, …]
    - Event list: [event time, event type, possibly other attributes of the event]

Approaches to Storing Lists in a Computer:

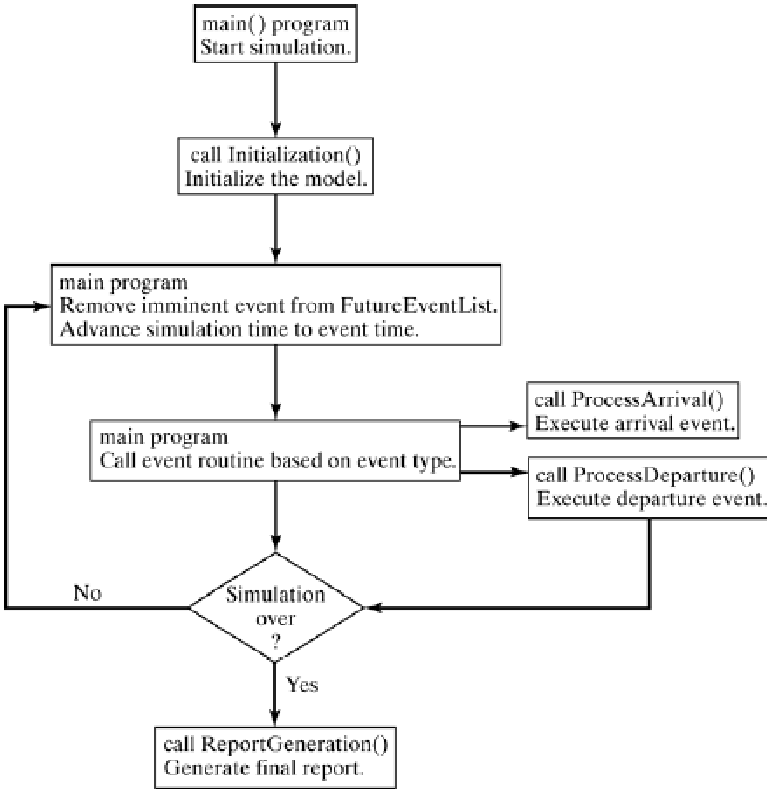
- *Sequential* allocation – approach used in Chap. 1
  - Records are in physically adjacent storage locations in the list, one record after another
  - Logical position = physical position
- *Linked* allocation
  - Logical location need not be the same as physical location
  - Each record contains its usual attributes, plus *pointers* (or *links*)
    - *Successor link* (or *front pointer*) – physical location (row number) of the record that's logically next in the list
    - *Predecessor link* (or *back pointer*) – physical location of the record that's logically before this one in the list
  - Each list has head pointer, tail pointer giving physical location of (logically) first and last records

Advantages of linked over sequential allocation

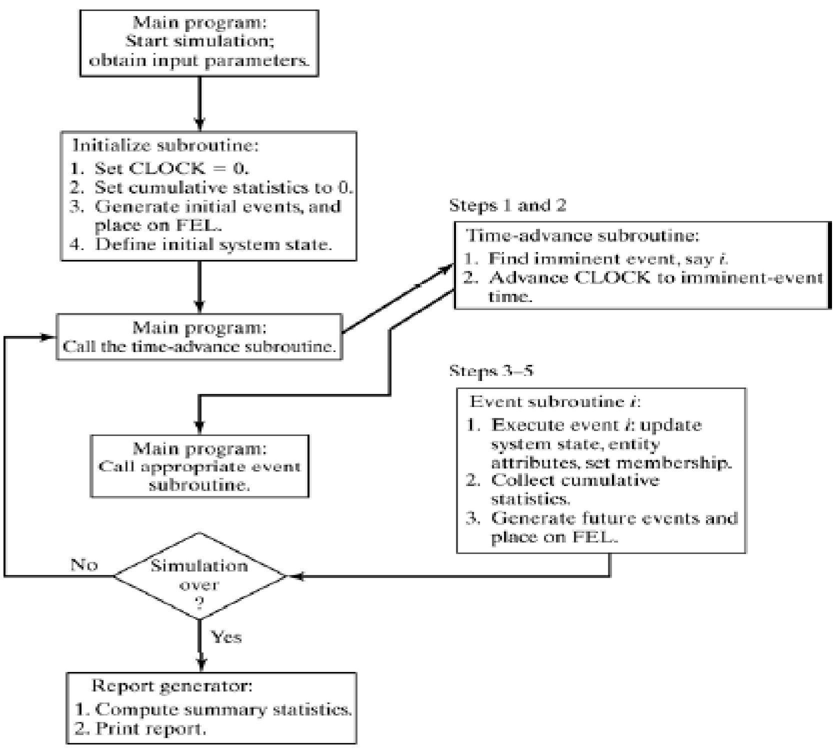
- Adding, deleting, inserting, moving records involves far fewer operations, so is much faster … critical for event-list management
  - Sequential allocation – have to move records around physically, copying all the attributes
  - Linked allocation – just readjust a few pointers, leave the record and attribute data physically where they are
- Reduce memory requirements without increasing chance of list overflow
  - Multiple lists can occupy the same physical storage area … can grow and shrink more flexibly than if they have their own storage area
- Provides a general modeling framework for list processing, which composes a lot of the modeling, computing in many simulations

3(b)Simuation In java

Overall structure of the Java program



Overall structure of an event-scheduling simulation program



```
class Sim {
    // Class Sim variables
    public static double clock,
                        meanInterArrivalTime,
                        meanServiceTime,
                        lastEventTime,
                        totalBusy,
                        maxQueueLength,
                        sumResponseTime;

    public static long  numberOfCustomers,
                        queueLength,
                        numberInService,
                        totalCustomers,
                        numberOfDepartures,
                        longService;

    public final static int arrival = 1;         // Event type for an arrival
    public final static int departure = 2;       // Event type for a departure

    public static EventList futureEventList;
    public static Queue customers;
    public static Random stream;

}
```

```
public static void main(String argv[]) {
        meanInterArrivalTime = 4.5;
        meanServiceTime      = 3.2;
        totalCustomers       = 1000;
        long seed            = Long.parseLong(argv[0]);

        stream = new Random(seed);           // Initialize rng stream
        futureEventList = new EventList();
        customers = new Queue();

        initialization();

        // Loop until first "totalCustomers" have departed
        while( numberOfDepartures < totalCustomers ) {
            Event event = (Event)futureEventList.getMin(); // Get imminent event
            futureEventList.dequeue();                     // Be rid of it
            clock = event.getTime();                       // Advance simulation time
            if( event.getType() == arrival ) {
                processArrival(event);
            }
            else {
                processDeparture(event);
            }
        }

        reportGeneration();
    }
```

4 (a) Newsdealer problem:

$$\text{Profit} = \begin{pmatrix}\text{revenue}\\ \text{from sales}\end{pmatrix} - \begin{pmatrix}\text{cost of}\\ \text{newspapers}\end{pmatrix} - \begin{pmatrix}\text{lost profit from}\\ \text{excess demand}\end{pmatrix} + \begin{pmatrix}\text{salvage from sale}\\ \text{of scrap papers}\end{pmatrix}$$

Random-Digit Assignments for Newspapers Demanded

	Cumulative Distribution			*Random-Digit Assignment*		
Demand	*Good*	*Fair*	*Poor*	*Good*	*Fair*	*Poor*
40	0.03	0.10	0.44	01–03	01–10	01–44
50	0.08	0.28	0.66	04–08	11–28	45–66
60	0.23	0.68	0.82	09–23	29–68	67–82
70	0.43	0.88	0.94	24–43	69–88	83–94
80	0.78	0.96	1.00	44–78	89–96	95–00
90	0.93	1.00	1.00	79–93	97–00	
100	1.00	1.00	1.00	94–00		

Table 2.18 Simulation Table for Purchase of 70 Newspapers

Day	*Random Digits for Type of Newsday*	*Type of Newsday*	*Random Digits for Demand*	*Demand*	*Revenue from Sales*	*Lost Profit from Excess Demand*	*Salvage from Sale of Scrap*	*Daily Profit*
1	94	Poor	80	60	$30.00	–	$0.50	$7.40
2	77	Fair	20	50	25.00	–	1.00	2.90
3	49	Fair	15	50	25.00	–	1.00	2.90
4	45	Fair	88	70	35.00	–	–	11.90
5	43	Fair	98	90	35.00	$3.40	–	8.50
6	32	Good	65	80	35.00	1.70	–	10.20
7	49	Fair	86	70	35.00	–	–	11.90
8	00	Poor	73	60	30.00	–	0.50	7.40
9	16	Good	24	70	35.00	–	–	11.90
10	24	Good	60	80	35.00	1.70	–	10.20

5 (a):

H_0: the random variable is Poisson distributed.
H_1: the random variable is not Poisson distributed.

x_i	Observed Frequency, O_i		Expected Frequency, E_i		$(O_i - E_i)^2/E_i$
0	12	22	2.6	12.2	7.87
1	10		9.6		
2	19		17.4		0.15
3	17		21.1		0.8
4	19		19.2		4.41
5	6		14.0		2.57
6	7		8.5		0.26
7	5	17	4.4	7.6	11.62
8	5		2.0		
9	3		0.8		
10	3		0.3		
> 11	1		0.1		
	100		100.0		27.68

$E_i = n \cdot p(x) = n \cdot \frac{e^{-\alpha}\alpha^x}{x!}$

Combined because of the assumption of $\min E_i = 5$, *e.g.*, $E_1 = 2.6 < 5$, *hence combine with* E_2

- Degree of freedom is k-s-1 = 7-1-1 = 5, hence, the hypothesis is rejected at the α=0.05 level of significance.

$$\chi_0^2 = 27.68 > \chi^2_{0.05,5} = 11.1$$

(b)

The goodness of fit (GOF) tests measure the compatibility of a random sample with a theoretical probability distribution function. In other words, these tests show how well the distribution you selected fits to your data.

The general procedure consists of defining a test statistic which is some function of the data measuring the distance between the hypothesis and the data, and then calculating the proba bility of obtaining data which have a still larger value of this test statistic than the value observed, assuming the hypothesis is true. This probability is called the confidence level.

Small probabilities (say, less than one percent) indicate a poor fit. Especially high probabilities (close to one) correspond to a fit which is too good to happen very often, and may indicate a mistake in the way the test was applied.

Kolmogorov-Smirnov Test

This test is used to decide if a sample comes from a hypothesized continuous distributuion. It is based on the empirical cumulative distribution function (ECDF). Assume that we have a random sample x_1, ... , x_n from some continuous distribution with CDF *F(x)*. The empirical CDF is denoted by

$$F_n(x) = \frac{1}{n} \cdot \left[\text{Number of observations} \leq x\right]$$

Definition

The Kolmogorov-Smirnov statistic (D) is based on the largest vertical difference between *F(x)* and $F_n(x)$. It is defined as

$$D_n = \sup_x |F_n(x) - F(x)|$$

H_0: The data follow the specified distribution.
H_A: The data do not follow the specified distribution.

The hypothesis regarding the distributional form is rejected at the chosen significance level (*alpha*) if the test statistic, D, is greater than the critical value obtained from a table.

Chi-Squared Test

The Chi-Squared test is used to determine if a sample comes from a population with a specific distribution. This test is applied to binned data, so the value of the test statistic depends on how the data is binned.

Although there is no optimal choice for the number of bins (k), there are several formulas which can be used to calculate this number based on the sample size (N). For example, EasyFit employs the following empirical formula:

$$k = 1 + \log_2 N$$

The data can be grouped into intervals of *equal probability* or *equal width*. The first approach is generally more acceptable since it handles peaked data much better. Each bin should contain at least 5 or more data points, so certain adjacent bins sometimes need to be joined together for this condition to be satisfied.

Definition

The Chi-Squared statistic is defined as

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i},$$

where O_i is the observed frequency for bin i, and E_i is the expected frequency for bin i calculated by

$$E_i = F(x_2) - F(x_1),$$

where F is the CDF of the probability distribution being tested, and x_1, x_2 are the limits for bin i.

H_0: The data follow the specified distribution.
H_A: The data do not follow the specified distribution.

The hypothesis regarding the distributional form is rejected at the chosen significance level (α) if the test statistic is greater than the critical value defined as

$$\chi^2_{1-\alpha,\, k-1}$$

meaning the Chi-Squared inverse CDF with k-1 degrees of freedom and a significance level of α.

6(a):

Simulation Table for Queueing Problem

A Customer	B *Time Since Last Arrival (Minutes)*	C *Arrival Time*	D *Service Time (Minutes)*	E *Time Service Begins*	F *Time Customer Waits in Queue (Minutes)*	G *Time Service Ends*	H *Time Customer Spends in System (Minutes)*	I *Idle Time of Server (Minutes)*
1	–	0	4	0	0	4	4	0
2	8	8	1	8	0	9	1	4
3	6	14	4	14	0	18	4	5
4	1	15	3	18	3	21	6	0
5	8	23	2	23	0	25	2	2
6	3	26	4	26	0	30	4	1
7	8	34	5	34	0	39	5	4
8	7	41	4	41	0	45	4	2
9	2	43	5	45	2	50	7	0
10	3	46	3	50	4	53	7	0

1. The average waiting time for a customer is 2.8 minutes. This is determined in the following manner:

$$\text{average waiting time (minutes)} = \frac{\text{total time customers wait in queue (minutes)}}{\text{total numbers of customers}} = \frac{56}{20} = 2.8 \text{ minutes}$$

2. The probability that a customer has to wait in the queue is 0.65. This is determined in the following manner:

$$\text{probability (wait)} = \frac{\text{number of customers who wait}}{\text{total number of customers}} = \frac{13}{20} = 0.65$$

The fraction of idle time of the server is 0.21. This is determined in the following manner:

$$\text{probability of idle server} = \frac{\text{total idle time of server (minutes)}}{\text{total run time of simulation (minutes)}} = \frac{18}{86} = 0.21$$

)

7 a)

The quantile-quantile (q-q) plot is a graphical technique for determining if two data sets come from populations with a common distribution.

A q-q plot is a plot of the quantiles of the first data set against the quantiles of the second data set. By a quantile, we mean the fraction (or percent) of points below the given value. That is, the 0.3 (or 30%) quantile is the point at which 30% percent of the data fall below and 70% fall above that value.

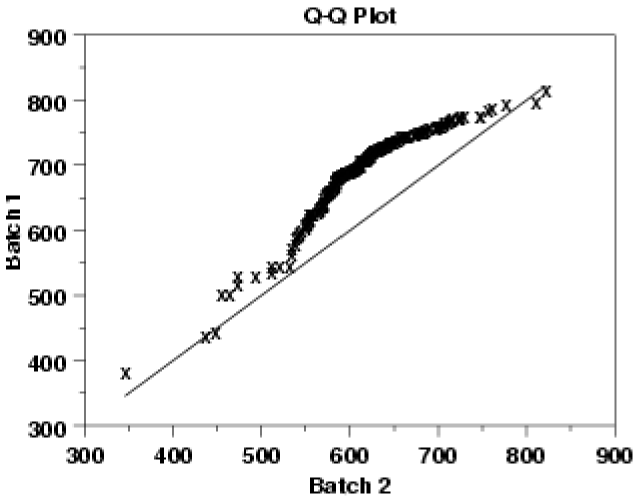
A 45-degree reference line is also plotted. If the two sets come from a population with the same distribution, the points should fall approximately along this reference line. The greater the departure from this reference line, the greater the evidence for the conclusion that the two data sets have come from populations with different distributions.

The advantages of the q-q plot are:

1. The sample sizes do not need to be equal.
2. Many distributional aspects can be simultaneously tested. For example, shifts in location, shifts in scale, changes in symmetry, and the presence of outliers can all be detected from this plot. For example, if the two data sets come from populations whose distributions differ only by a shift in location, the points should lie along a straight line that is displaced either up or down from the 45-degree reference line.

The q-q plot is similar to a probability plot. For a probability plot, the quantiles for one of the data samples are replaced with the quantiles of a theoretical distribution.

Sample Plot



The q-q plot is formed by:

- Vertical axis: Estimated quantiles from data set 1
- Horizontal axis: Estimated quantiles from data set 2

Both axes are in units of their respective data sets. That is, the actual quantile level is not plotted. For a given point on the q-q plot, we know that the quantile level is the same for both points, but not what that quantile level actually is.

b**) Input Modeling Process:**

Steps in input modeling:

1) Collect data from real system of interest
   - Requires substantial time and effort
   - Use expert opinion in case of no sufficient data
2) Identify a probability distribution to represent the input process
   - Draw frequency distribution, histograms
   - Choose a family of theoretical distribution
3) Estimate the parameters of the selected distribution
4) Apply goodness-of-fit tests to evaluate the chosen distribution and the parameters
   - Chi-square tests
   - Kolmogorov Smirnov Tests
5) If these tests are not justified, choose a new theoretical distribution and go to step 3! If all theoretical distributions fail, then either use emprical distribution or recollect data.

Step1: Data Collection:

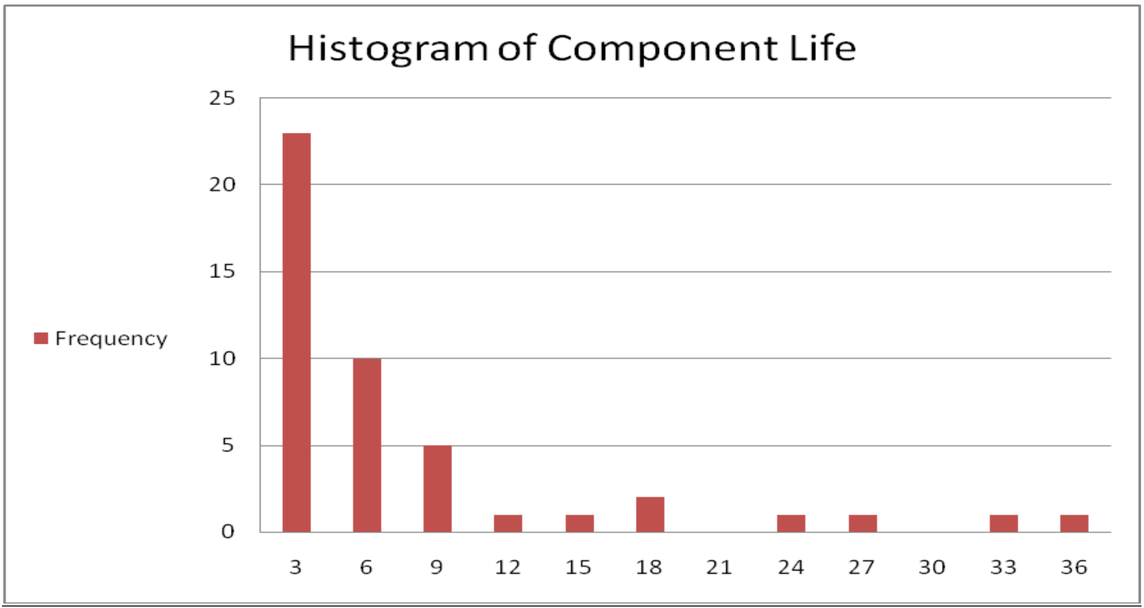
- Nonhomogeneous interarrival time distribution; distribution changes with time of the day, days of the week, etc. You can't merge all these data for distribution fitting!
- Two arrival processes might be dependent; like demand for washing machines and dryers. You shouldn't treat them seperately!
- Start and end of service durations might not be clear; You should split the service into well defined processes!
- Machines may breakdown randomly; You should collect data for up and down times!

Step 2: Identify the Probability Distribution

- Histogram with Continuous Data :Using raw data

Component Life (days)	Frequency
[0-3)	23
[3-6)	10
[6,9)	5
[9-12)	1
[12-15)	1
[15-18)	2
[18-21)	0
[21-24)	1
[24-27)	1
[27-30)	0
[30-33)	1
[33-36)	1

79.919	3.081	0.062	1.961	5.845
3.027	6.505	0.021	0.013	0.123
6.769	59.899	1.192	34.760	5.009
18.387	0.141	43.565	24.420	0.433
144.695	2.663	17.967	0.091	9.003
0.941	0.878	3.148	2.157	7.579
0.624	5.380	3.371	7.078	23.960
0.590	1.928	0.300	0.002	0.543
7.004	31.764	1.005	1.147	0.219
3.217	14.382	1.008	2.336	4.562



Step 3: Estimate the parameters of the selected distribution

- Sample mean and the sample variance are the point estimators for the population mean and population variance

  Let X_i; $i=1,2,\dots,n$ iid random variables (raw data are known) , then the sample mean and sample variance s^2 are calculated as

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \quad s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1} = \frac{\sum_{i=1}^{n} X_i^2 - n\bar{X}^2}{n-1}$$

- If the data are discrete and have been grouped in a frequency distribution, i.e., the raw data are not known, then

  where *k* is the number of distinct values of *X* and f_j; $j=1,2,...,k$ is the observed frequency of the value X_j of *X*.

$$\bar{X} = \frac{\sum_{j=1}^{k} f_j X_j}{n} \quad s^2 = \frac{\sum_{j=1}^{k} f_j X_j^2 - n\bar{X}^2}{n-1}$$

- The minimum, mod (i.e., data value with the highest frequency) and maximum of the population data are estimated from the sample data as

Step 4: Goodness of fit test

- Goodness of fit tests (GFTs) provide helpful guidance for evaluating the suitability of the selected input model as a simulation input.
- GFTs check the discrepancy between the emprical and the selected theoretical distribution to decide whether the sample is taken from that theoretical distribution or not.
- The role of sample size, *n*:
  - If n is small, GFTs are unlikely to reject any theoretical distribution, since discrepancy is attributed to the sampling error!
  - If n is large, then GFTs are likely to reject almost all distributions.
- Chi square test is valid for large sample sizes and for both discrete and continuous assumptions when parameters are estimated with maximum likelihood.
- **Hypothesis test:**

  Ho: The random variable X conforms to the theoretical distribution with the estimated parameters

  Ha: The random variable does NOT conform to the theoretical distribution with the estimated parameters

We need a test statistic to either reject or fail to reject Ho. This test statistic should measure the discrepency between the theoretical and the emprical distribution.

If this test statistic is high, then Ho is rejected,

Otherwise we fail to reject Ho! (Hence we accept Ho)

Test statistic:

Arrange n observations into a set of k class intervals or cells. The test statistic is given by

$$\chi_0^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

where O_i is the observed frequency in the i^{th} class interval and

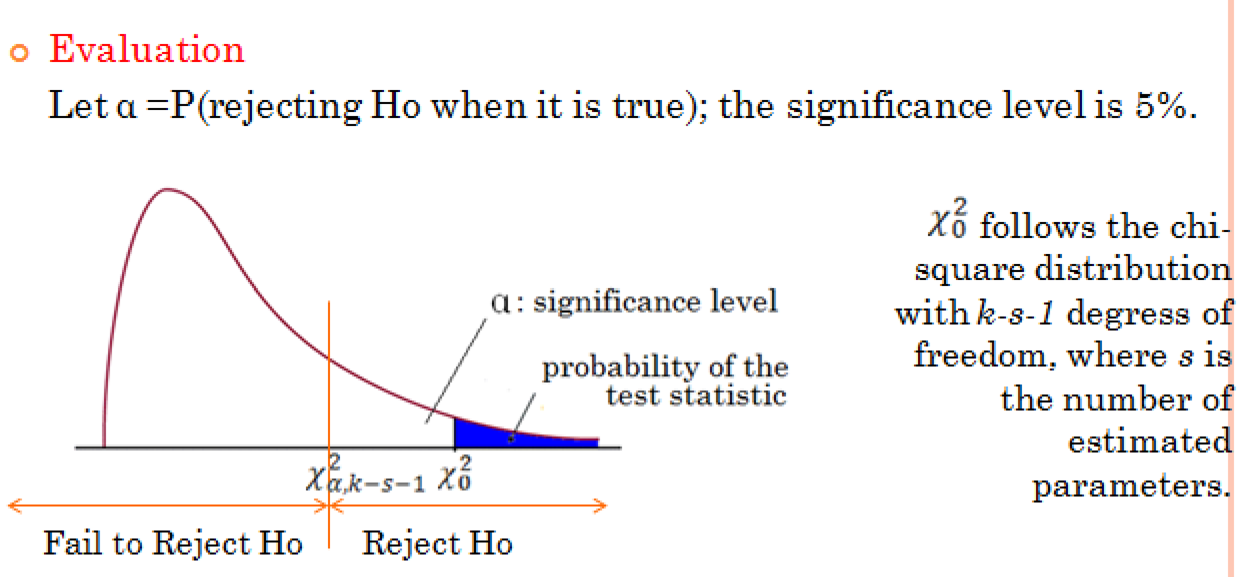
E_i is the expected frequency in the i^{th} class interval.

$$E_i = np_i$$

where p_i is the theoretical probability associated with the i^{th} class, i.e., p_i =P(random variable X belongs to i^{th} class).

- Evaluation

  Let α =P(rejecting Ho when it is true); the significance level is 5%.



If probability of the test statistic < α, reject Ho and the distribution

otherwise, fail to reject Ho.

8 a) Write the characteristics of queueing systems and Queueing notations

- Simulation is often used in the analysis of queuing models. In a simple but typical queuing model, shown in Figure 6.1, customers arrive from time to time and join a queue (waiting line), are eventually served, and finally leave the system.
- The term "customer" refers to any type of entity that can be viewed as requesting "service" from a system.
- Therefore, many service facilities, production systems, repair and maintenance facilities, communications and computer systems, and transport and material-handling systems can be viewed as queuing systems.

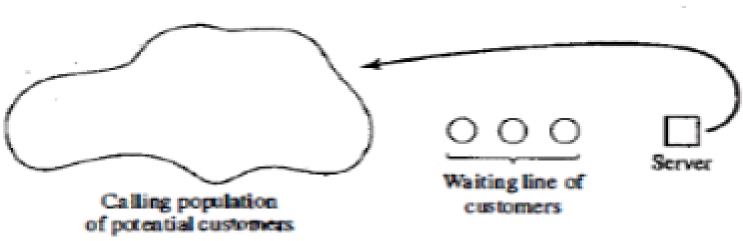


Figure 6.1 Simple queueing model

The Calling Population

- The population of potential customers, referred to as the calling population, may be assumed to be finite or infinite.

- For example, consider a bank of five machines that are curing tires. After an interval of time, a machine automatically opens and must be attended by a worker who removes the tire and puts an uncured tire into the machine.
- The machines are the "customers;' who "arrive" at the instant they automatically open. The worker is the "server," who "serves" an open machine as soon as possible.
- The calling population is finite and consists of the five machines.
- In systems with a large population of potential customers, the calling population is usually assumed to be infinite. For such systems, this assumption is usually innocuous and, furthermore, it might simplify the model.
- Examples of infinite populations include the potential customers of a restaurant, bank, or other similar service facility and also very large groups of machines serviced by a technician.

System Capacity

- In any queuing systems, there is a limit to the number of customers that may be in the waiting line or system.
- For example, an automatic car wash might have room for only 10 cars to waiting the line to enter the mechanism.
- It might be too dangerous (or illegal) for cars to wait in the street. An arriving customer who finds the system full does not enter but returns immediately to the calling population.
- Some systems, such as concert ticket sales for students, may be considered as having unlimited capacity, since there are no limits on the number of students allowed to wait 'to purchase tickets.
- As will be seen later, when a system has limited capacity, a distinction is made between the arrival rate (i.e., the number of arrivals per time unit) and the effective arrival rate (i.e., the number who arrive and enter the system per time unit).

The Arrival Process

- The arrival process for infinite-population models is usually characterized in terms of interarrival times of successive customers.
- Arrivals may occur at scheduled times or at random times. When at random times, the inter arrival times are usually characterized by a probability distribution. In addition, customers may arrive one at a time or in batches.
- The batch may be of constant size or of random size.
- One important application of finite population models is the machine-repair problem. The machines are the customers, and a runtime is also called time to failure.
- When a machine fails, it "arrives" at the queuing system (the repair facility) and remains there until it is "served" (repaired).

Queue Behavior and Queue Discipline

- Queue behavior refers to the actions of customers while in a queue waiting for service to begin.
- In some situations, there is a possibility that incoming customers will balk (leave when they see that the line is too long), renege (leave after being in the line when they see that the line is moving too slowly), or jockey (move from one line to another if they think they have chosen a slow line).
- Queue discipline refers to the logical ordering of customers in a queue and determines which customer will be chosen for service when a server becomes free. OR

- Queue discipline refers to the rule that a server uses to choose the next customer from the queue when the server completes the service of the current customer.
- Common queue disciplines include first-in-first-out (FIFO); last-in-first-out (LIFO); service in random order (SIRO); shortest processing time first (SPT); and service according to priority (PR).

1. First in first out :This principle states that customers are served one at a time and that the customer that has been waiting the longest is served first.
2. Last in first out : This principle also serves customers one at a time, however the customer with the shortest waiting time will be served first. Also known as a stack.
3. Processor sharing: Service capacity is shared equally between customers.
4. Priority : Customers with high priority are served first.[17] Priority queues can be of two types,
5. Non-pre emptive (where a job in service cannot be interrupted) and pre emptive (where a job in service can be interrupted by a higher priority job). No work is lost in either model.
6. Shortest job first: The next job to be served is the one with the smallest size
7. Pre emptive shortest job first:The next job to be served is the one with the original smallest size.
8. Shortest remaining processing time: The next job to serve is the one with the smallest remaining processing requirement.

Service Times and the Service Mechanism

- The service times of successive arrivals are denoted by S1S1, S2S2, S3S3, . . . They may be constant or of random duration.
- In the latter case, {S1S1, S2S2, S3S3, . . . }is usually characterized as a sequence of independent and identically distributed random variables.
- The exponential, Weibull, gamma, lognormal and truncated normal distributions have all been used successfully as models of service times in different situations. Sometimes services are identically distributed for all customers of a given type or class or priority, whereas customers of different types might have completely different service-time distributions.
- In addition, in some systems, service times depend upon the time of day or upon the length of the waiting line. For example, servers might work faster than usual when the waiting line is long, thus effectively reducing the service time.

Queuing discipline

The queuing discipline describes the order in which arrivals are serviced. Common queue disciplines include FIFO, shorter service time first, and random selection for the nest service. The queuing discipline also includes characteristics of the system such as maximum queue-length) when the queue reaches this maximum, arrivals turn away or balk) and customer reneging (customer waiting in line become impatient and leave the system before service)

Classification of queuing system models

A/Bs/K/E where A: specifies the arrival process
B: specifics the services process
s: specific's the number of servers
K: maximum number of customers allowed into the system
E: queue discipline

Symbols

M: exponentially-distributed service or arrival times
D: constant service or arrival times

Measures of performance of a congestion systems:

Ls: Expected number of customers in the system
Lq: Expected number of customers in the queue
Ws: Expected time a customer is in the system, including the time for service
Wq: Expected time a customer waits for service

M/M/1 queuing model

Queuing models are most easily developed when the arrival times and service times are exponentially distributed. Although the assumption of exponentially-distributed arrival and service times may seem unrealistic, this group of models has wide application and can also serve as a useful first pass in the analysis of more complex congestion systems.

8 b)Able Baker Problem:

7a)