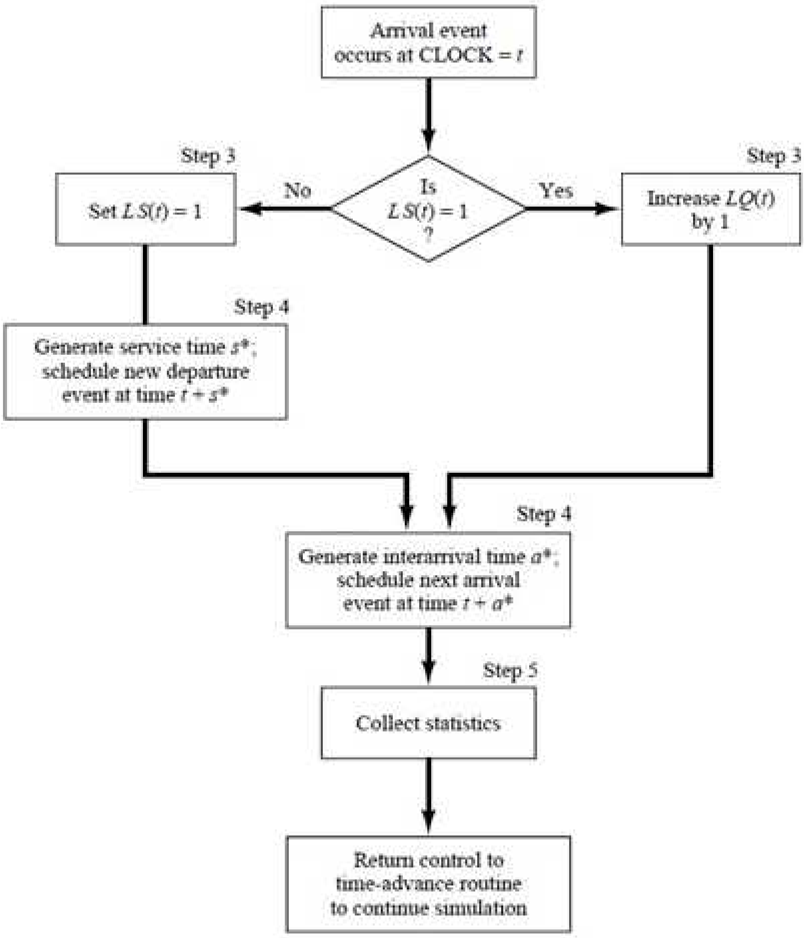


Figure Execution of the arrival event.

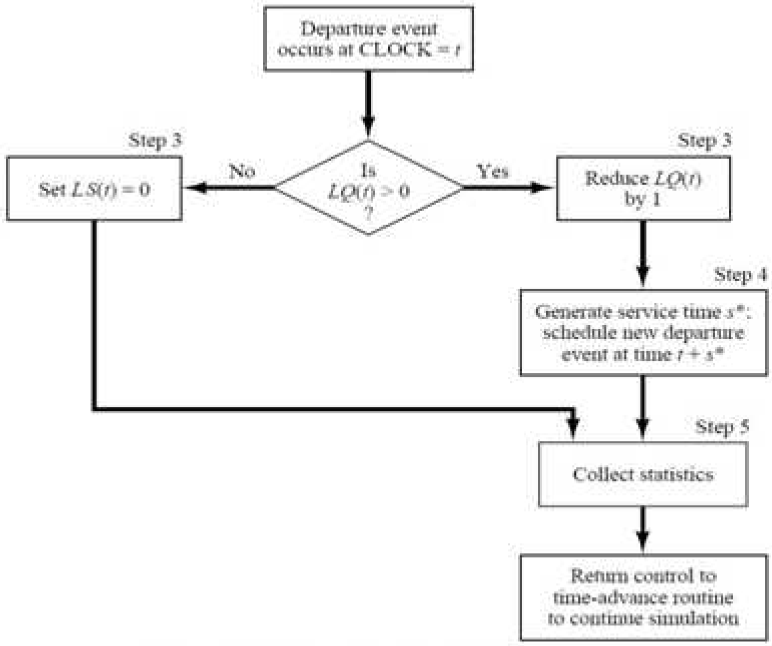


Figure Execution of the departure event.

4.List Processing

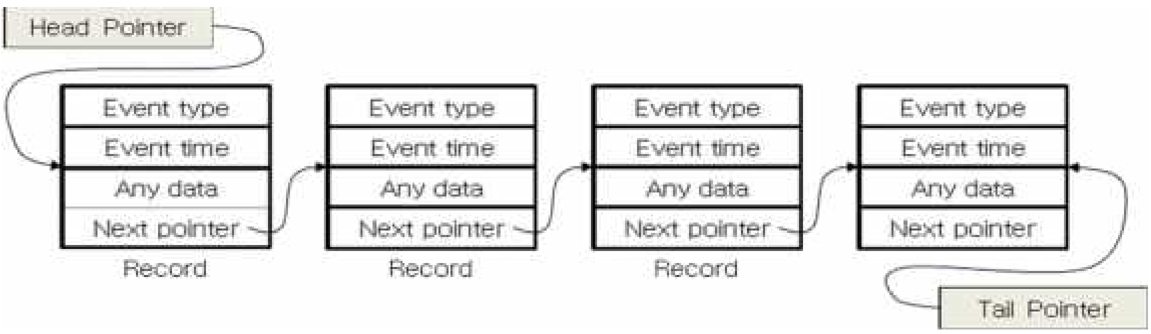
List processing deals with methods for handling lists of entities and the future event list

Basic properties and Operations

1. They have a head pointer/ top pointer
2. Some even have tail pointer

Operations

1. Removing a record from the top of the list
2. Removing a record from any location on the list
3. Adding an entity record to the top or bottom of the list
4. Adding a record to an arbitrary position on the list, determined by the ranking rule.



1. **Removing a record from any location on the list.**
   - If an arbitrary event is being canceled, or an entity is removed from a list based on some of its attributes (say, for example, its priority and due date) to begin an activity.
   - By making a partial search through the list.

2. **Adding an entity record to the top or bottom of the list.**
   - When an entity joins the back of a first-in first-out queue.
   - by adjusting the tail pointer on the FEL by adding an entity to the bottom of the FEL
3. **Adding a record to an arbitrary position on the list, determined by the ranking rule.**
   - If a queue has a ranking rule of earliest due date first (EDF).
   - By making a partial search through the list.

The goal of list-processing techniques: to make second and fourth operations efficient

- **The notation R(i) :** the i^{th} record in the array
- **Advantage:** Any specified record, say the i^{th}, can be retrieved quickly without searching, merely by referencing R (i).
- **Disadvantage:** When items are added to the middle of a list or the list must be rearranged.
  - o Arrays typically have a fixed size, determined at compile time or upon initial allocation when a program first begins to execute.
  - o In simulation, the maximum number of records for any list may be difficult or impossible to determine ahead of time, while the current number in a list may vary widely over the course of the simulation run.

5.Simulation in java

- Java is a widely used programming that has been used extensively in simulation.
- The following components are common to almost all models written in java
- **Clock:** a variable defining simulated time
- **Initialization method: a** method to define the system state at time 0.
- **Min-time event methods:** a method that identifies the imminent event, that is the element of the future event list that has the smallest time-stamp
- **Event methods:** for each even type, a method to update system state when that event occurs
- **Random-variate generators** methods to generate samples from desired probability distributions
- **Main program :**to maintain overall control of the event –scheduling algorithm
- **Report generator:** a method that computes summary statistics from cumulative statistics and prints a report at the end of the simulation

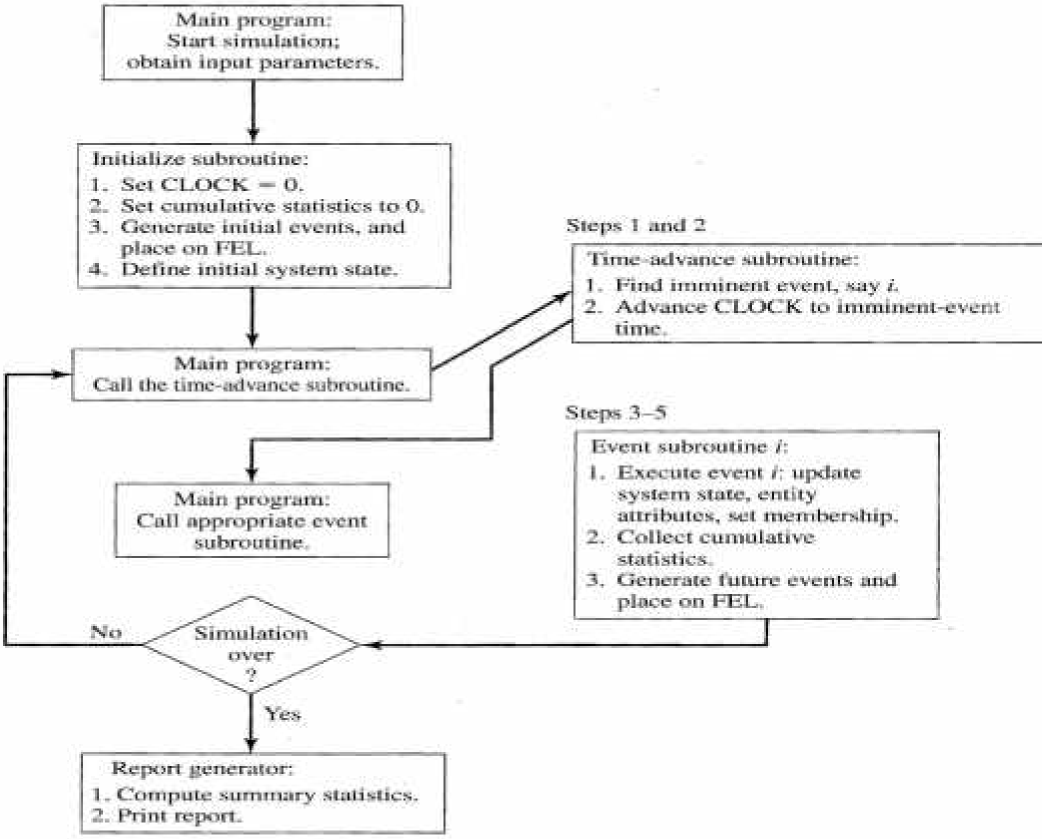


Figure 4.1 Overall structure of an event-scheduling simulation program.

Single server queue simulation in java

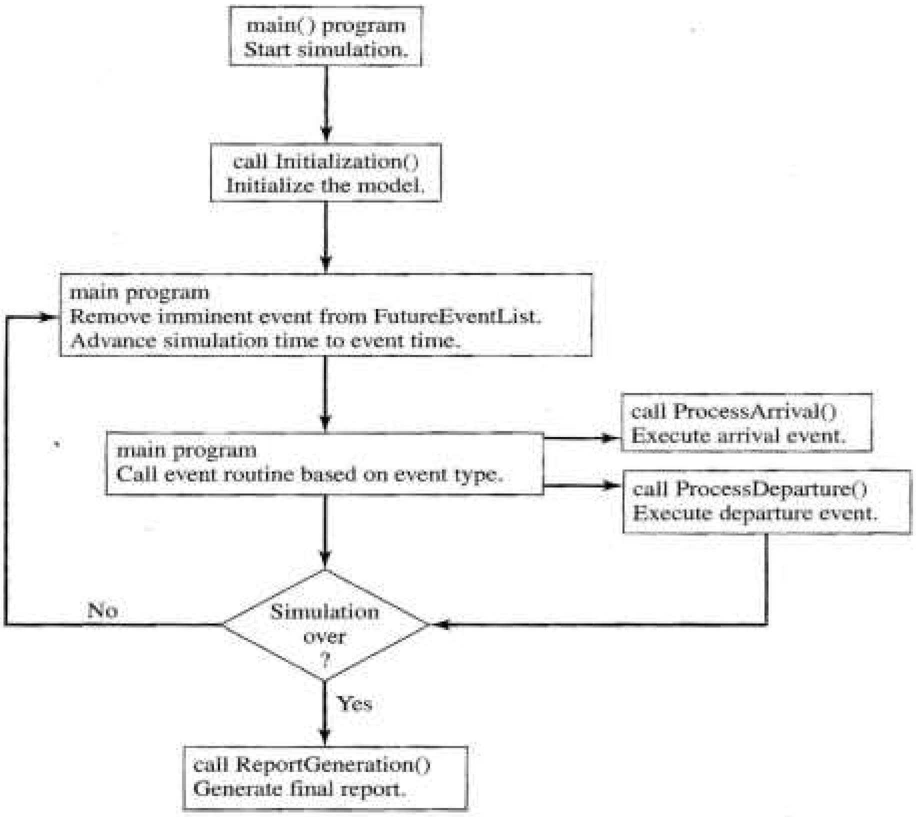


Figure 4.2 Overall structure of Java simulation of a single-server queue.

Simulation in GPSS:

- **Generate block**-represents the arrival event with the IAT time specified by RVEXPO(1,&IAT).
- **QUEUE-** is to work in conjunction with the depart block to collect data on queue or any subsystem.
- **Queue block with line-begins** data collection for the waiting line before the cashier.
- **Size block** with checkout-once truncation representing a customer captures the cashier represented by the resource checkout the data collection for the waiting line ends.
- **Advance block with RVNORM**-random number stream 1 is being used; the mean time for normal distribution is given by ampervariable &MEAN .
- **CHECKOUT with RELEASE** block-the end of the data collection for response times is indicated by the DEPART block for the queue SYSTEM.
- **MI GE 400 WITH TEST**- its check weather customer spend more than 4min in system, if the customer spent more than 4min count is incremented otherwise its terminate the task.

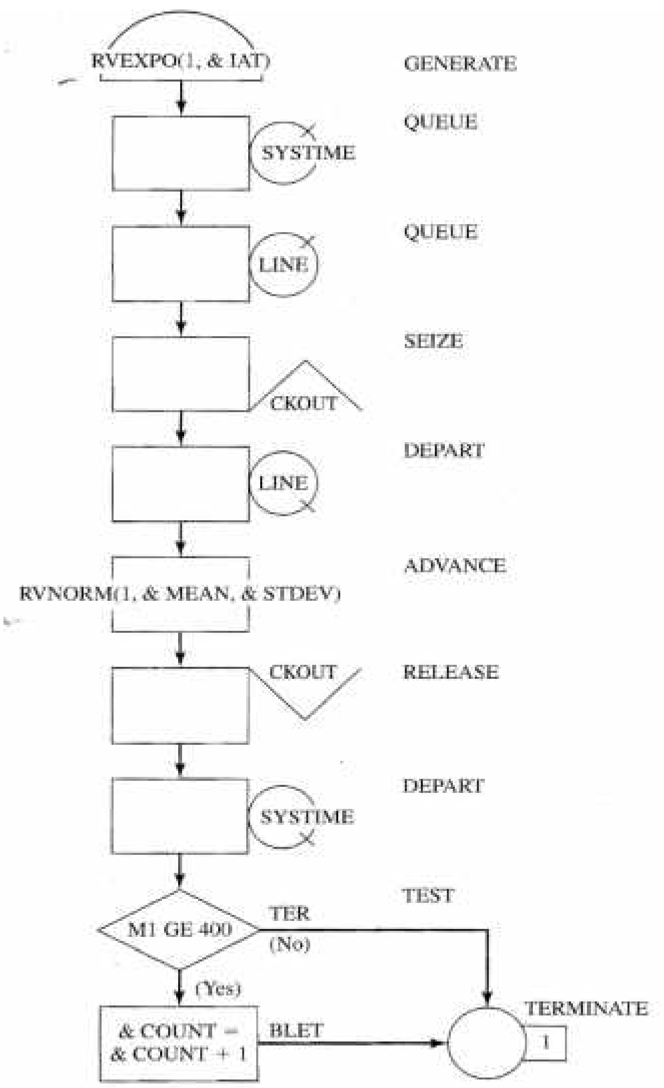


Figure 4.10 GPSS block diagram for the single-server queue simulation.

(1) UNIT-2 [NOTE: Check out time column should be included in prob. No 1 to 5. It is included in prob. 6 to 8.]

Problems on Event Scheduling / Time advanced Algorithm

1. Prepare a simulation table for 1 channel queuing s/m using ES/TA algorithm. Stopping event is at 30.

IAT : 8 6 1 8 2 8

ST : 4 1 4 3 2 4

LQ → Load Queue

LS → Load Service

Step 1 : Compute Departure Time

C. No	Inter Arrival Time	Arrival Time	Service Time	Departure Time
1	–	0	4	4
2	8	8	1	9 (8+1)
3	6	14	4	18 (14+4)
4	1	15	3	21 (18+3)
5	8	23	2	25 (23+2)
6	2	25	4	29 (25+4)
7	8	33	–	–

Step 2 : Simulation Table for 6 customers

Event Type	Clock	System State		Future Event List	Cumulative Statistics	
		LQ(E)	LS(E)		Busy	Maximum Queue
A_1	0	0	1	(D, 4) (A, 8) (E, 30)	0	0
D_1	4	0	0	(A, 8) (D, 9) (E, 30)	4	0
A_2	8	0	1	(D, 9) (A, 14) (E, 30)	4	0
D_2	9	0	0	(A, 14) (D, 18) (E, 30)	5	0
A_3	14	0	1	(A, 15) (D, 18) (E, 30)	5	0
A_4	15	1	1	(D, 18) (A, 23) (E, 30)	6	1
D_3	18	0	1	(D, 21) (A, 23) (E, 30)	9	1
D_4	21	0	0	(A, 23) (D, 25) (E, 30)	12	1
A_5	23	0	1	(D, 25) (A, 25) (E, 30)	12	1

A_6/D_5	25	0	1	(A, 33) (D, 29) (E, 30)	14	1
D_6	29	0	0	(E, 30) (A, 33)	14	1
~~D_6~~	~~29~~	~~0~~	~~0~~	~~(E, 30)~~	~~18~~	~~1~~

2. Stopping Time = 30

IAT : 2 4 3 1 2 5 6

ST : 4 3 2 5 2 1

i] Compute Departure Time

C. No	IAT	AT	ST	DT
1	–	0	4	4
2	2	2	3	7
3	4	6	2	9
4	3	9	5	14
5	1	10	2	16
6	2	12	1	17
7	5	17	–	–
8	6	23	–	–

ii] Simulation Table for 6 customers

Event Type	Clock	System State		Future Event List	CS	
		LQ(t)	LS(t)		B	MQ
A_1	0	0	1	(A, 2) (D, 4) (E, 30)	0	0
A_2	2	1	1	(D, 4) (A, 6) (E, 30)	2	1
D_1	4	0	1	(A, 6) (D, 7) (E, 30)	4	1
A_3	6	1	1	(D, 7) (A, 9) (E, 30)	6	1
D_2	7	0	1	(A, 9) (D, 9) (E, 30)	7	1
A_4/D_3	9	0	1	(A, 10) (D, 14) (E, 30)	9	1
~~A_4~~	[illegible]	[illegible]	[illegible]	~~(A, 10) (D, 14) (E, 30)~~	[illegible]	[illegible]

2

A_6	12	2	1	(D, 14) (A, 17) (E, 30)	12	2
D_4	14	1	1	(A, 17) (D, 16) (E, 30)	14	2
D_5	16	1	1	(A, 17) (D, 17) (E, 30)	16	2
A_7/D_6	17	0	1	(A, 23) (E, 30)	17	2

3. Stopping Time = 60

IAT : 1 1 6 3 7 5 2 4 1

ST : 4 2 5 4 1 5 4 1 4

i) Compute Departure Time

C.No	IAT	AT	ST	DT
1	–	0	4	4
2	1	1	2	6
3	1	2	5	11
4	6	8	4	15
5	3	11	1	16
6	7	18	5	23
7	5	23	4	27
8	2	25	1	28
9	4	29	4	33
10	1	–	–	–

ii) Simulation Table for 9 customers

Event Type	Clock	S/m State LQ(t)	S/m State LS(t)	Future Event List	CS B	CS MQ
A_1	0	0	1	(A, 1) (D, 4) (E, 60)	0	0
A_2	1	1	1	(A, 2) (D, 4) (E, 60)	1	1
A_3	2	2	1	(D, 4) (A, 8) (E, 60)	2	2
D_1	4	1	1	(D, 6) (A, 8) (E, 60)	4	2
D_2	6	0	1	(A, 8) (D, 11) (E, 60)	6	2
A_4	8	1	1	(A, 11) (D, 11) (E, 60)	8	2

D_3/A_5	11	1	1	(D,15) (A,18) (E,60)	11	2
~~D_3~~				~~(D,15) (A,18) (E,60)~~		
D_4	15	0	1	(D,16) (A,18) (E,60)	15	2
D_5	16	0	0	(A,18) (D,23) (E,60)	16	2
A_6	18	0	1	(D,23) (A,23) (E,60)	16	2
D_6	23	0	1	(A,23) (D,27) (E,60)	21	2
A_7	23	0	1	(D,27) (A,25) (E,60)	21	2
A_8	25	1	1	(D,27) (A,29) (E,60)	23	2
D_7	27	0	1	(A,29) (A,28) (E,60)	25	2
D_8	28	0	0	(A,29) (D,33) (E,60)	26	2
A_9	29	0	1	(D,33) (E,60)	26	2
D_9	33	0	0	(E,60)	30	2

4. Prepare a simulation table using ES/TA algorithm until the clock reaches time 23 using IAT & ST given :
Stopping Time is 30.

IAT : 5 1 2 3 4 9 5 8 6 1

ST : 4 7 8 1 4 2 5 3 1 4

i) Compute Departure Time

C. No	IAT	AT	ST	DT
1	–	0	4	4
2	5	5	7	12
3	1	6	8	20
4	2	8	1	21
5	3	11	4	25
6	4	15	2	27
7	9	24	5	32
8	5	29	3	35

C. No	IAT	AT	ST	DT
9	8	37	1	38
10	6	43	4	47
11	1	–	–	–

(3) ii] Simulation Table for 9 customers.

Event Type	Clock	s/m state		Future Event List	CS	
		$LQ(t)$	$LS(t)$		B	MQ
A_1	0	0	1	(A, 1) (D, 4)	0	0
A_2	1	1	1	(D, 4) (A, 7)	1	1
D_1	4	0	1	(D, 6) (A, 7)	4	1
D_2	6	0	0	(A, 7) (D, 12)	6	1
A_3	7	0	1	(A, 8) (D, 12)	6	1
A_4	8	1	1	(A, 11) (D, 12)	7	1
A_5	11	2	1	(D, 12) (A, 16)	10	2
D_3	12	1	1	(A, 16) (D, 17)	11	2
A_6	16	2	1	(D, 17) (A, 23)	15	2
D_4	17	1	1	(D, 18) (A, 23)	16	2
D_5	18	0	1	(D, 22) (A, 23)	17	2
D_6	22	0	0	(A, 23) (D, 27)	21	2
A_7	23	0	1	(A, 24) (D, 27)	21	2
A_8	24	1	1	(D, 27) (A, 28)	22	2
D_7	27	0	1	(A, 28) (D, 31)	25	2
A_9	28	1	1	(D, 31) (D, 32)	26	2
D_8	31	0	1	(D, 32)	29	2
D_9	32	1	0	–	30	2

S → Customer who taken response time from system
N_D → No. of customer departured
F → customer who spent more than given time (Eg: 4 min)
S = Response Time + Current Departure
Response Time = Clock − Current Arrival

ii) Simulation Table

Event Type	Clock	S/m state LQ(t)	LS(t)	Future Event List	CS B	MQ
A_1	0	0	1	(D, 4) (A, 5) (E, 30)	0	0
D_1	4	0	0	(A, 5) (D, 12) (E, 30)	4	0
A_2	5	0	1	(A, 6) (D, 12) (E, 30)	4	0
A_3	6	1	1	(A, 8) (D, 12) (E, 30)	5	1
A_4	8	2	1	(A, 11) (D, 12) (E, 30)	7	2
A_5	11	3	1	(D, 12) (A, 15) (E, 30)	10	3
D_2	12	2	1	(A, 15) (D, 20) (E, 30)	11	3
A_6	15	3	1	(D, 20) (A, 24) (E, 30)	14	3
D_3	20	2	1	(D, 21) (A, 24) (E, 30)	19	3
D_4	21	1	1	(A, 24) (D, 25) (E, 30)	20	3

5. Prepare a simulation table using Time Advanced Algorithm:

IAT : 1 6 1 3 5 7 1 4

ST : 4 2 5 5 1 4 4 4 1

i) Compute Departure Time

C. No	IAT	AT	ST	DT
1	–	0	4	4
2	1	1	2	6
3	6	7	5	12
4	1	8	5	17
5	3	11	1	18
6	5	16	4	22
7	7	23	4	27
8	1	24	4	31
9	4	28	1	32

6. Prepare a simulation table using Time advanced algorithm (4) with IAT : 1 1 8 6 8

ST : 4 2 3 4 1 2

Find customers who spent more than 4 min.

i) Compute Departure Time

C. No	IAT	AT	ST	DT
1	–	0	4	4
2	1	1	2	6
3	1	2	3	9
4	8	10	4	14
5	6	16	1	17
6	8	24	2	26

ii) Simulation table of 6 customers.

Event type	Clock	S/m state		Check out time	Future Event List	CS				
		$LQ(t)$	$LS(t)$			S	N_D	F	B	MQ
A_1	0	0	1	$(C_1, 0)$	(A, 1) (D, 4)	0	0	0	0	0
A_2	1	1	1	$(C_1, 0)$ $(C_2, 1)$	(A, 2) (D, 4)	0	0	0	1	1
A_3	2	2	1	$(C_1, 0)$ $(C_2, 1)$ $(C_3, 2)$	(D, 4) (A, 10)	0	0	0	2	2
D_1	4	1	1	$(C_2, 1)$ $(C_3, 2)$	(D, 6) (A, 10)	4	1	0	4	2
D_2	6	0	1	$(C_3, 2)$	(D, 9) (A, 10)	9	2	1	6	2
D_3	9	0	0	–	(A, 10) (D, 14)	16	3	2	9	2
A_4	10	0	1	$(C_4, 10)$	(D, 14) (A, 16)	16	3	2	9	2
D_4	14	0	0	–	(A, 16) (D, 17)	20	4	2	13	2
A_5	16	0	1	$(C_5, 16)$	(D, 17) (A, 24)	20	4	2	13	2
D_5	17	0	0	–	(A, 24) (D, 26)	21	5	2	14	2
A_6	24	0	1	$(C_6, 24)$	(D, 26)	21	5	2	14	2
D_6	26	0	0	–	–	23	6	2	16	2

7. Consider single server queue with one checkout counter using ES/TA algorithm

IAT : 4 2 8 1 8 3 6 8

ST : 4 6 5 2 3 4 4 1

Find the no. of customers who spent 4 or more ~~min~~ in the system. Stopping time = 32

i) Compute Arrival & Departure time

C.NO	IAT	AT	ST	DT
1	–	0	4	4
2	4	4	6	10
3	2	6	5	15
4	8	14	2	17
5	1	15	3	20
6	8	23	4	27
7	3	26	4	31
8	6	32	1	33
9	8	40	–	–

ii) Simulation table

Event type	clock	S/m state LQ(t)	LS(t)	check out time	Future Event List	CS S	N_D	F	B	MQ
A_1	0	0	1	$(C_1,1)$	(A,4) (D,4) (E,32)	0	0	0	0	0
D_1/A_2	4	0	1	$(C_2,4)$	(A,6) (D,10) (E,32)	4	1	1	4	0
A_3	6	1	1	$(C_2,4)$ $(C_3,6)$	(D,10) (A,14) (E,32)	4	1	1	6	1
D_2	10	0	1	$(C_3,6)$	(A,14) (D,15) (E,32)	10	2	2	10	1
A_4	14	1	1	$(C_3,6)$ $(C_4,14)$	(A,15) (D,15) (E,32)	10	2	2	14	1
A_5/D_3	15	1	1	$(C_4,14)$ $(C_5,15)$	(D,17) (A,23) (E,32)	19	3	3	15	1
D_4	17	0	1	$(C_5,15)$	(D,20) (A,23) (E,32)	22	4	3	17	1
D_5	20	0	0	-	(A,23) (D,27) (E,32)	27	5	4	20	1
A_6	23	0	1	$(C_6,23)$	(A,26) (D,27) (E,32)	27	5	4	20	1

5	D_6	27	0	1	$(C_7, 26)$	(D,31) (A,32) (E,32)	31	6	5	24	1
	D_7	31	0	0	–	(A,32) (D,33) (E,32)	36	7	6	28	1
	A_8	32	0	1	$(C_8, 32)$	(A,40) (D,33) (E,32)	36	7	6	28	1

8. Develop a simulation table for single server queue with one check out counter using TA algorithm. Find busy time of server, maximum queue length, Total no. of customer who spent 3 min or more in system, Total number of departure.

IAT : 1 6 8 8 3 8 4 2 8

ST : 4 1 4 4 2 3 5 6 4

i) Compute Arrival and Departure time

C.No	IAT	AT	ST	DT
1	–	0	4	4
2	1	1	1	5
3	6	7	4	11
4	8	15	4	19
5	8	23	2	25
6	3	26	3	29
7	8	34	5	39
8	4	38	6	45
9	2	40	4	49
10	8	48	–	–

Event type	Clock	S/m state LQ(t)	LS(t)	Check Out Time	Future Event List	CS S	N_D	F	B	MQ
A_1	0	0	1	$(C_1, 0)$	(A, 1) (D, 4)	0	0	0	0	0
A_2	1	1	1	$(C_1, 0)(C_2, 1)$	(A, 7) (D, 4)	0	0	0	1	1
D_1	4	0	1	$(C_2, 1)$	(A, 7) (D, 5)	4	1	1	4	1
D_2	5	0	0	–	(A, 7) (D, 11)	8	2	2	5	1
A_3	7	0	1	$(C_3, 7)$	(A, 15) (D, 11)	8	2	2	5	1
D_3	11	0	0	–	(A, 15) (D, 19)	12	3	3	9	1
A_4	15	0	1	$(C_4, 15)$	(D, 19) (A, 23)	12	3	3	9	1
D_4	19	0	0	–	(A, 23) (D, 25)	16	4	4	13	1
A_5	23	0	1	$(C_5, 23)$	(D, 25) (A, 26)	16	4	4	13	1
D_5	25	0	0	–	(A, 26) (D, 29)	18	5	4	15	1
A_6	26	0	1	$(C_6, 26)$	(D, 29) (A, 34)	18	5	4	15	1
D_6	29	0	0	–	(A, 34) (D, 39)	21	6	5	18	1
A_7	34	0	1	$(C_7, 34)$	(A, 38) (D, 39)	21	6	5	18	1
A_8	38	1	1	$(C_7, 34)(C_8, 38)$	(D, 39) (A, 40)	21	6	5	22	1
D_7	39	0	1	$(C_8, 38)$	(A, 40) (D, 45)	26	7	6	23	1
A_9	40	1	1	$(C_8, 38)(C_9, 40)$	~~(A, 48)~~ (D, 45)	26	7	6	24	1
D_8	45	0	1	$(C_9, 40)$	~~(D, 48)~~ (D, 49)	33	8	7	29	1
[illegible]	~~48~~	[illegible]	[illegible]	~~$(C_9, 40)(C_{10}, 48)$~~	~~(D, 49)~~	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
D_9	49	0	0	$(C_{10}, 48)$	–	42	9	8	33	1

Busy time of server = 33 min

Maximum Queue Length = 1

Total no. of customer who spent 3 or more in System = 8

Total no. of departure = 9