UNIT 2: General Principles

1. Discrete-event simulation

- The basic building blocks of all discrete-event simulation models: entities and attributes, activities and events.
- A system is modeled in terms of
  - o Its state at each point in time
  - o The entities that pass through the system and the entities that represent system resources
  - o The activities and events that cause system state to change.
- Discrete-event models are appropriate for those systems for which changes in system state occur only at discrete points in time.
- This chapter deals exclusively with dynamic, stochastic systems (i.e., involving time and containing random elements) which change in a discrete manner.

1.1 Concepts in Discrete-Event Simulation(components of discrete event Simulation)

1. **System:** A collection of entities (e.g., people and machines) that together over time to accomplish one or more goals.
2. **Model:** An abstract representation of a system, usually containing structural, logical, or mathematical relationships which describe a system in terms of state, entities and their attributes, sets, processes, events, activities, and delays.
3. **System state:** A collection of variables that contain all the information necessary to describe the system at any time.
4. **Entity:** Any object or component in the system which requires explicit representation in the model (e.g., a server, a customer, a machine).
5. **Attributes:** The properties of a given entity (e.g., the priority of a v customer, the routing of a job through a job shop).
6. **List:** A collection of (permanently or temporarily) associated entities ordered in some logical fashion (such as all customers currently in a waiting line, ordered by first come, first served, or by priority).
7. **Event:** An instantaneous occurrence that changes the state of a system as an arrival of a new customer).
8. **Event notice:** A record of an event to occur at the current or some future time, along with any associated data necessary to execute the event; at a minimum, the record includes the event type and the event time.
9. **Event list:** A list of event notices for future events, ordered by time of occurrence; also known as

the future event list (FEL).

10. **Activity:** A duration of time of specified length (e.g., a service time or arrival time), which is known when it begins (although it may be defined in terms of a statistical distribution).
11. **Delay:** A duration of time of unspecified indefinite length, which is not known until it ends (e.g., a customer's delay in a last-in, first-out waiting line which, when it begins, depends on future arrivals).
12. **Clock:** A variable representing simulated time.

1.2 The Event-Scheduling/Time-Advance Algorithm

- **The mechanism for advancing simulation time and guaranteeing that all events occur in correct chronological order is based on the future event list (FEL).**
- **Future Event List (FEL)**
  - **To contain all event notices for events that have been scheduled to occur at a future time.**
  - **To be ordered by event time,** meaning that the events are arranged chronologically; that is, the event times satisfy.
  - **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 the end-activity event, together with its event time, is placed on the future event list.

The sequence of actions which a simulator must perform to advance the clock system snapshot is called the event- scheduling/time-advance algorithm.

The system snapshot at time t=0 and t=t1 (VIP VTU question)

ClK	System State		Future Event List	
T	(5,1,6)		(3, t1)— Type 3 event to occur at timet1 (1, t2)— Type 1 event to occur at time t2 (1, t3)- Type 1 event to occur at time t_3 (2, tn)— Type 2 event to occur at time t_n	

Event-scheduling/time-advance algorithm

Step 1. Remove the event notice for the imminent event

(event 3, time t\) from PEL

Step 2. Advance CLOCK to imminent event time

(i.e., advance CLOCK from r to t1).

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 PEL ranked by event time. (Example:

Event 4 to occur at time t*, where t2 < t* < t3.)

Step 5. Update cumulative statistics and counters.

<u>New system snapshot at time *t1*</u>

OCK	System		Future Event List	
T1	(5,1,5)		(1, t2)— Type 1 event to occur at time t1 (4, t*)— Type 4 event to occur at time t* (1, t3)— Type 1 event to occur at time t3 (2, tn)— Type 2 event to occur at time tn	

1.3 World Views

- During simulation, a modeler adopts a **<u>world view or orientation for developing a model</u>**.
- Those most prevalent are **the event scheduling world view, the process-interaction worldview, and the activity-scanning world view.**

1. **The process-interaction approach**, a simulation analyst thinks in terms of processes.
   - õ The **process-interaction approach** is popular because of its intuitive appeal, and because the simulation packages that implement it allow an analyst to describe the process flow in terms of high-level block or network constructs.
   - õ Figure 3.4 shows the interaction between two customer processes as customer n+1 is delayed until the previous customer's “end-service event” occurs.

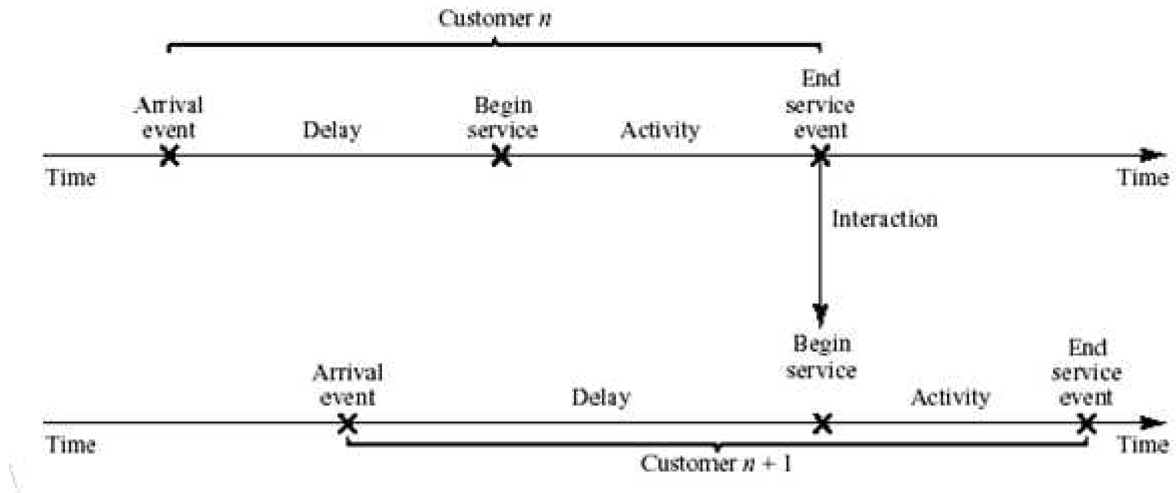


fig: Two interacting customer process in a single server queue

2. When using the event-scheduling approach, a simulation analyst concentrates on events and their effect on system state. Both the event-scheduling and the process-interaction approaches use a variable time advance.
3. **The activity-scanning approach** uses a fixed time increment and a rule-based approach to decide whether any activities can begin at each point in simulated time.

- **The pure activity scanning approach** has been modified by what is called the **three-phase approach.**
- In the three-phase approach, events are considered to be activity duration-zero time units. With this definition, activities are divided into two categories called B and C.
- **B activities:** Activities bound to occur; all primary events and unconditional activities.
- **C activities:** Activities or events those are conditional upon certain conditions being true.

With the three-phase approach the simulation proceeds with repeated execution of the three phases until it is completed:

1. **Phase A:** Remove the imminent event from the FEL and advance the clock to its event time. Remove any other events from the FEL that have the event time.
2. **Phase B:** Execute all B-type events that were removed from the FEL.
3. **Phase C:** Scan the conditions that trigger each C-type activity and activate any whose conditions are met. Rescan until no additional C-type activities can begin or events occur.

2.Manual Simulation Using Event Scheduling

In an event-scheduling simulation, a simulation table is used to record the successive system snapshots as time advances.

Let us consider the example of a grocery shop which has only one checkout counter. **(Single-Channel Queue)**

The system consists of those customers in the waiting line plus the one (if any) checking out. The model has the following components:

System state (LQ (t), LS (t)), where LQ (t) is the number of customers in the waiting line, and LS (t) is the number being served (0 or 1) at time t.

Entities: The server and customers are not explicitly modeled, except in terms of the state variables above.

Events

Arrival(A)

Departure(D)

Stopping event (E), scheduled to occur at time 60.

Event notices

(A, t). Representing an arrival event to occur at future time t (D, t), representing a customer departure at future time t

(E, 60), representing the simulation-stop event at future time 60

Activities

Interarrival time, Service time,

Delay Customer time spent in waiting line.

In this model, the FEL will always contain either two or three event notices.

Flow Chart for execution of arrival and departure event using time advance /Event scheduling algorithm (vtu Question)

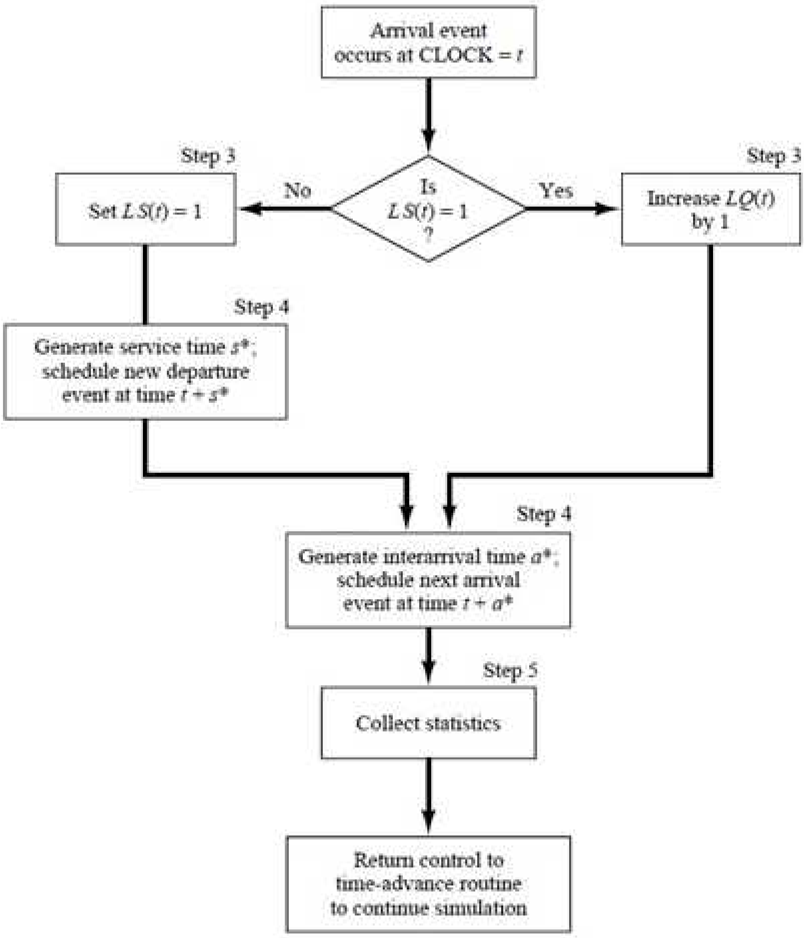


Figure Execution of the arrival event.

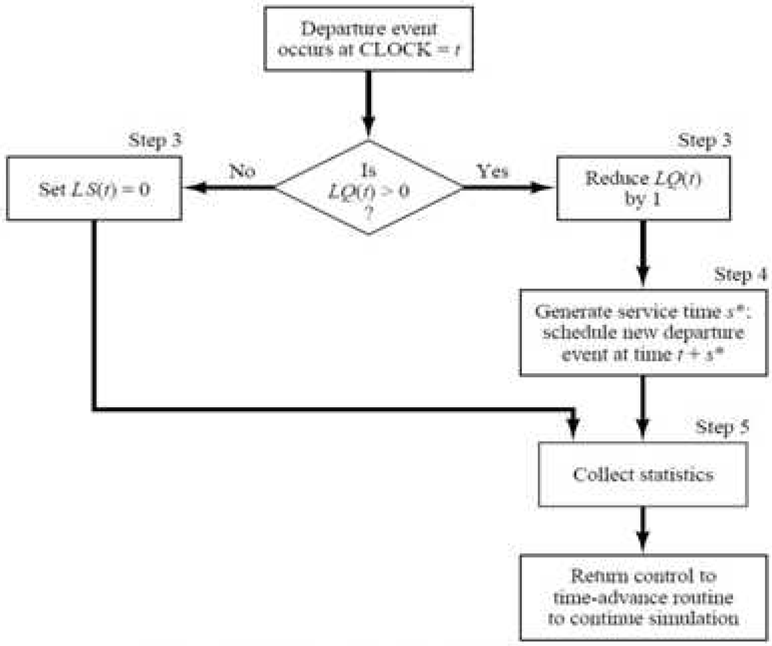


Figure Execution of the departure event.

4.List Processing

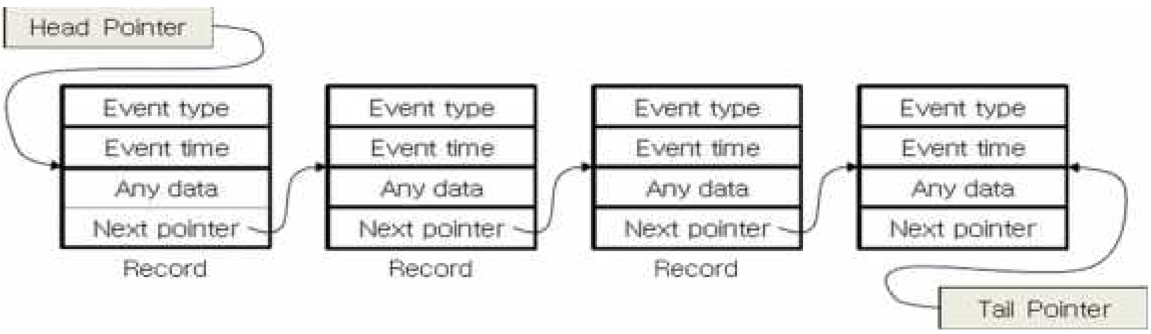
List processing deals with methods for handling lists of entities and the future event list

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.
    - Arrays typically have a fixed size, determined at compile time or upon initial allocation when a program first begins to execute.
    - 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

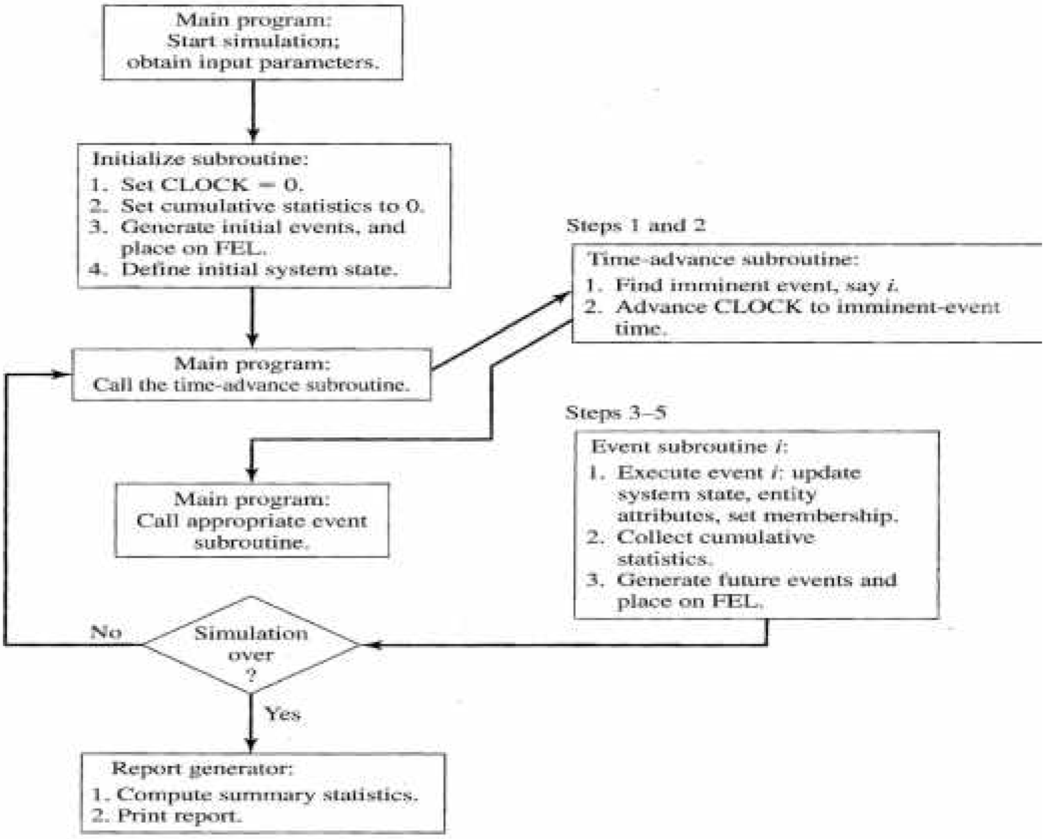


Figure 4.1 Overall structure of an event-scheduling simulation program.

Single server queue simulation in java

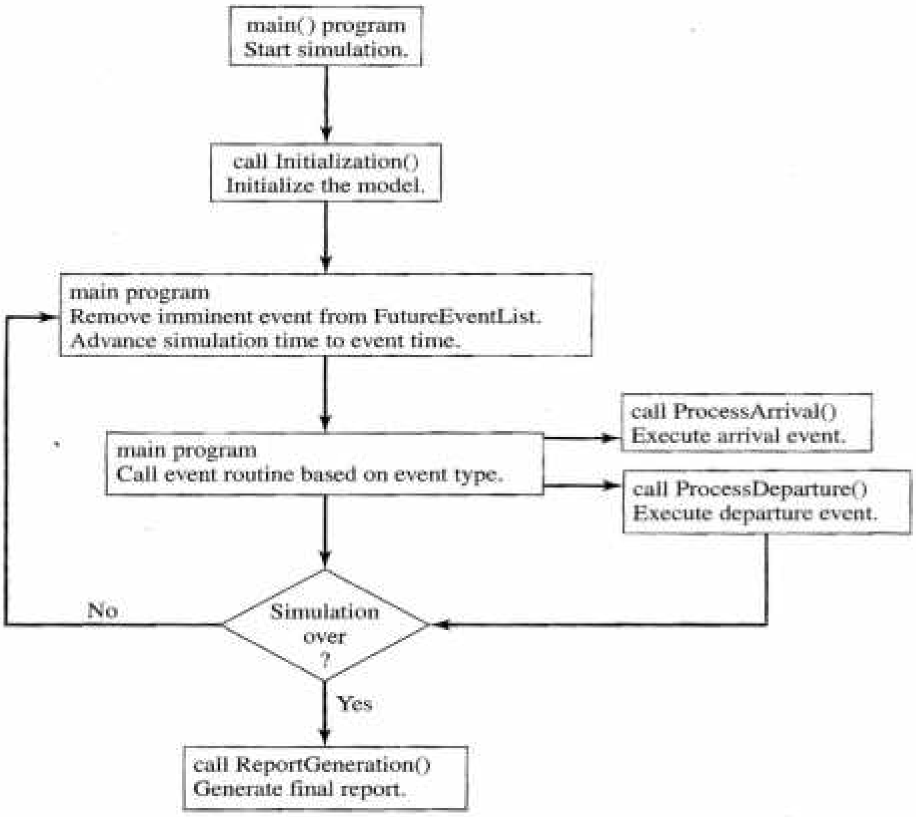


Figure 4.2 Overall structure of Java simulation of a single-server queue.

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.

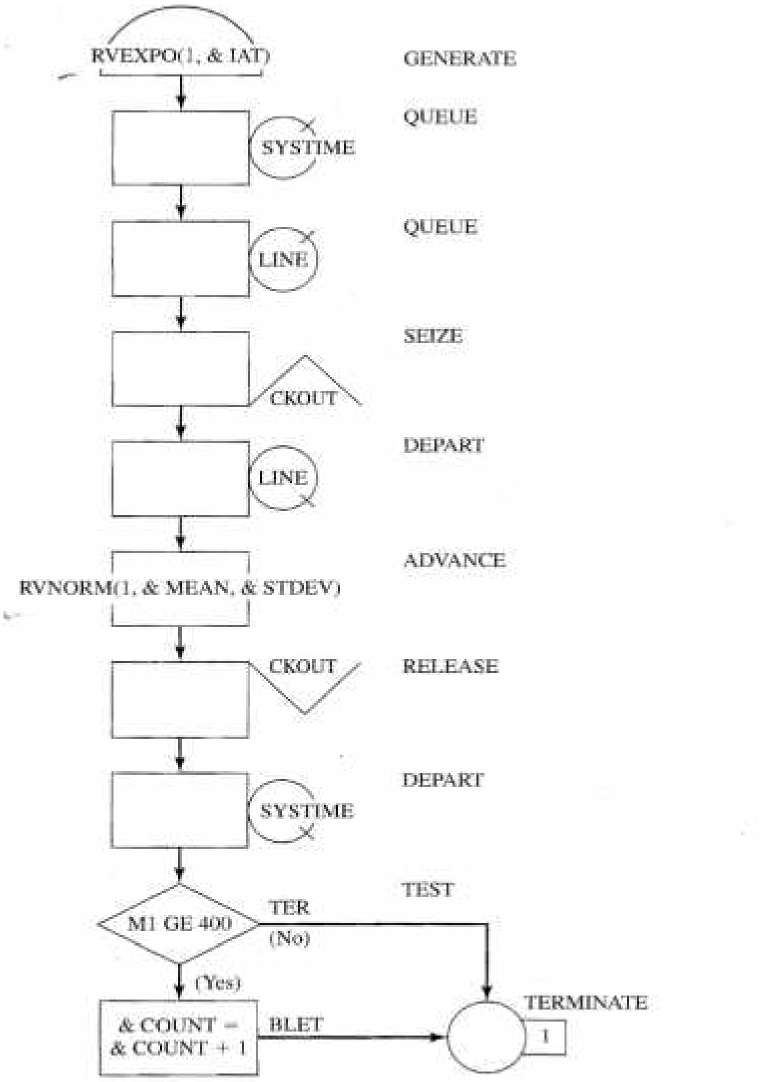


Figure 4.10 GPSS block diagram for the single-server queue simulation.

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~~	~~4~~

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)	LS(t)	Future Event List	CS B	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
[illegible]	[illegible]	[illegible]	[illegible]	~~(D,15) (A,18) (E,60)~~	[illegible]	[illegible]
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 with IAT : 1 1 8 6 8

(4)

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 ~~mi~~ 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		checkout time	Future Event List	CS				
		LQ(t)	LS(t)			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	$(C_9, 40)$	~~(A, 48)~~ (D, 49)	33	8	7	29	1
~~A_{10}~~	~~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

Travel Times as given below: Consider stopping time 32 clock cycle.
OR The stopping event will be completion of 2 weighings (2 AQs).

Loading Times	5	15	10	15	15	10	5
Weighing Times	17	15	15	15	15	12	
Travel Times	40	60	60	60	80		

Solution:

Simulation table for Dump-truck Operation.

Clock t	System State				Lists		Future Event List	Cumulative Statistics	
	LQ(t)	L(t)	WQ(t)	W(t)	Loader Queue	Weigh Queue		B_L	B_S
0	4	2	0	1	DT_4 DT_5 DT_6 DT_7	-	$(E_L, 5, DT_2)$* $(E_L, 15, DT_3)$ $(E_W, 17, DT_1)$	0	0
5	3	2	1	1	DT_5 DT_6 DT_7	DT_2	$(E_L, 15, DT_3)$* $(E_W, 17, DT_1)$ $(E_L, 15, DT_4)$ (5+10)	10 (5-0)×2+0	5 (5-0)+0
15	2	2	2	1	DT_6 DT_7	DT_2 DT_3	$(E_W, 17, DT_1)$ $(E_L, 15, DT_4)$* $(E_L, 30, DT_5)$ (15+15)	30 (15-5)×2+10	15 (15-5)+5
15	1	2	3	1	DT_7	DT_2 DT_3 DT_4	$(E_W, 17, DT_1)$* $(E_L, 30, DT_5)$ $(E_L, 30, DT_6)$	30	15
17	1	2	2	1	DT_7	DT_3 DT_4	$(E_L, 30, DT_5)$* $(E_L, 30, DT_6)$ $(A_Q, 57, DT_1)$ (17+40) $(E_W, 32, DT_2)$ (17+15)	34	17
30	0	2	3	1	-	DT_3 DT_4 DT_5	$(E_L, 30, DT_6)$* $(A_{QL}, 57, DT_1)$ $(E_W, 32, DT_2)$ $(E_L, 40, DT_7)$	60	30
30	0	1	4	1	-	DT_3 DT_4 DT_5 DT_6	$(A_{QL}, 57, DT_1)$ $(E_W, 32, DT_2)$* $(E_L, 40, DT_7)$	60	30
32	0	1	3	1	-	DT_4 DT_5 DT_6	$(A_{QL}, 57, DT_1)$ $(E_L, 40, DT_7)$ $(A_{QL}, 92, DT_2)$ $(E_W, 47, DT_3)$	62	32

2. Consider 6 Dump-trucks with loading times, weighing time & Traveling times are given below,

Loading Times	5	5	10	15	20	5	5
Weighing Times	12	15	20	12	15	15	
Travel Time	40	60	20	80			

- Until Clock cycle 52
- Calculate
  i) Avg loader utilization
  ii) Avg scale utilization.

Solution:

Simulation table for Dump truck operations.

Clock	System State				Lists		Future Event List	Cumulative Statistics	
t	LQ(t)	L(t)	WQ(t)	W(t)	Loader queue	Weigh queue		B_L	B_S
0	3	2	0	1	DT_4 DT_5 DT_6	–	$(E_L, 5, DT_2)$* $(E_L, 5, DT_3)$ $(E_W, 12, DT_1)$	0	0
5	2	2	1	1	DT_5 DT_6	DT_2	$(E_L, 5, DT_3)$* $(E_W, 12, DT_1)$ $(E_L, 15, DT_4)$	10	5
5	1	2	2	1	DT_6	DT_2 DT_3	$(E_W, 12, DT_1)$* $(E_L, 15, DT_4)$ $(E_L, 20, DT_5)$	10	5
12	1	2	1	1	DT_6	DT_3	$(E_L, 15, DT_4)$* $(E_L, 20, DT_5)$ $(A_{QL}, 52, DT_1)$ $(E_W, 27, DT_2)$	24	12
15	0	2	2	1	-	DT_3 DT_4	$(E_L, 20, DT_5)$* $(A_{QL}, 52, DT_1)$ $(E_W, 27, DT_2)$ $(E_L, 35, DT_6)$	30	15
20	0	1	3	1	-	DT_3 DT_4 DT_5	$(A_{QL}, 52, DT_1)$ $(E_W, 27, DT_2)$* $(E_L, 35, DT_6)$	40	20
27	0	1	2	1	-	DT_4 DT_5	$(A_{QL}, 52, DT_1)$ $(E_L, 35, DT_6)$* $(A_{QL}, 87, DT_2)$ $(E_W, 47, DT_3)$	47	27
35	0	0	3	1	-	DT_4 DT_5 DT_6	$(A_{QL}, 52, DT_1)$ $(A_{QL}, 87, DT_2)$ $(E_W, 47, DT_3)$*	55	35

Clock t	System State LQ(t)	L(t)	WQ(t)	W(t)	Lists Loader queue	Weigh queue	Future Event List	Cumulative Statistics B_L	B_S
47	0	0	2	1	–	DT_5 DT_6	$(AQL, 52, DT_1)$ * $(AQL, 87, DT_2)$ $(AQL, 67, DT_3)$ $(E_W, 59, DT_4)$	55	47
52	0	1	2	1	–	DT_5 DT_6	$(AQL, 87, DT_2)$ $(AQL, 67, DT_3)$ $(E_L, 57, DT_1)$ [52+5] $(E_W, 59, DT_4)$	55	52

i) Average loader utilization $= \frac{55/2}{52} = \underline{0.529}$

ii) Average scale utilization $= \frac{52}{52} = \underline{1.00}$

3. Consider Loading Times, weighing Times & Travel times of 6 Dump trucks. Loading Times & weighing Times are based on FIFO, until clock cycle 60 (or) entire system runs for 1 hours (or) estimation time is 60 mins.

Loading Times	10	10	15	5	10	5
Weighing Times	10	15	5	10	10	
Travel Times	40	60	80	100		

* if all the trucks have equal value use FIFO method.

Solution:

Simulation table for Dump-truck operation.

Clock t	System State LQ(t)	L(t)	WQ(t)	W(t)	Lists Loaders queue	Weigh queue	Future event List	Cumulative Statistics B_L	B_S
0	3	2	0	1	DT_4 DT_5 DT_6	–	$(E_L, 10, DT_2)$ $(E_L, 10, DT_3)$ $(E_W, 10, DT_1)$*	0	0
10	2	2	0	1	DT_5 DT_6	–	$(E_L, 10, DT_3)$* $(AQL, 50, DT_1)$ [10+40] $(E_W, 25, DT_2)$ [10+15] $(E_L, 25, DT_4)$ [10+15]	20	10
10	1	2	1	1	DT_6	DT_3	$(AQL, 50, DT_1)$ $(E_W, 25, DT_2)$ $(E_L, 25, DT_4)$ $(E_L, 15, DT_5)$* [10+5]	20	10

Clock t	System State				Lists		Future Event List	Cumulative Statistics	
	LQ(t)	L(t)	WQ(t)	W(t)	Loader Queue	Weigh Queue		B_L	B_S
15	0	2	2	1	–	DT_3 DT_5	$(A_{QL}, 50, DT_1)$ $(E_W, 25, DT_2)$* $(E_L, 25, DT_4)$ $(E_L, 25, DT_6)$	30	15
25	0	2	1	1	–	DT_5	$(A_{QL}, 50, DT_1)$ $(E_L, 25, DT_4)$* $(E_L, 25, DT_6)$ $(A_{QL}, 85, DT_2)$ $(E_W, 30, DT_3)$	50	25
25	0	1	2	1	–	DT_5 DT_4	$(A_{QL}, 50, DT_1)$ $(E_L, 25, DT_6)$* $(A_{QL}, 85, DT_2)$ $(E_W, 30, DT_3)$	50	25
25	0	0	3	1	–	DT_5 DT_4 DT_6	$(A_{QL}, 50, DT_1)$ $(A_{QL}, 85, DT_2)$ $(E_W, 30, DT_3)$*	50	25
30	0	0	2	1	–	DT_4 DT_6	$(A_{QL}, 50, DT_1)$ $(A_{QL}, 85, DT_2)$ $(A_{QL}, 110, DT_3)$ $(E_W, 40, DT_5)$*	50	30
40	0	0	1	1	–	DT_6	$(A_{QL}, 50, DT_1)$* $(A_{QL}, 85, DT_2)$ $(A_{QL}, 110, DT_3)$ $(A_{QL}, 140, DT_5)$ $(E_W, 50, DT_4)$	50	40
50	0	1	1	1	–	DT_6	$(A_{QL}, 85, DT_2)$ $(A_{QL}, 110, DT_3)$ $(E_L, 55, DT_1)$ $(E_W, 50, DT_4)$* $(A_{QL}, 140, DT_5)$	50	50
50	0	1	0	1	–	–	$(A_{QL}, 85, DT_2)$ $(A_{QL}, 110, DT_3)$ $(A_{QL}, 140, DT_5)$ $(A_{QL}, 90, DT_4)$ $(E_W, 60, DT_6)$ $(E_L, 55, DT_1)$*	50	50
55	0	0	1	1	–	DT_1	$(A_{QL}, 85, DT_2)$ $(A_{QL}, 110, DT_3)$ $(A_{QL}, 140, DT_5)$ $(A_{QL}, 90, DT_4)$ $(E_W, 60, DT_6)$*	55	55

Clock	System State				Lists		Future Event	Cumulative Statistics	
t	$LQ(t)$	$L(t)$	$WQ(t)$	$W(t)$	Loader Queue	Weigh Queue	List	B_L	B_S
60	0	0	0	1	—	—	(ALQ, 85, DT_2) (ALQ, 110, DT_3) (ALQ, 140, DT_2) (ALQ, 90, DT_4) (ALQ, 120, DT_6) (EW, 70, DT_1)	55	60