Simulating a M/M/1 Queue

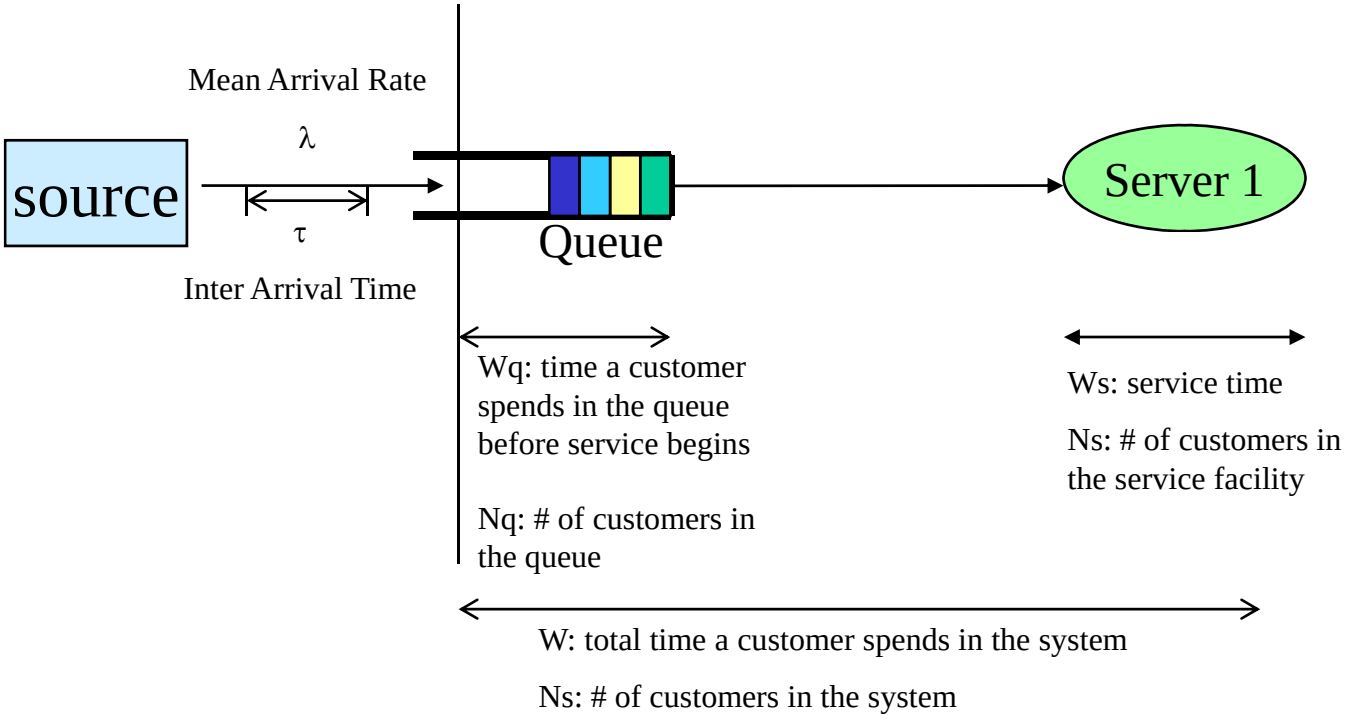
Chang-Gun Lee (cglee@snu.ac.kr)

Assistant Professor

The School of Computer Science and Engineering

Seoul National University

Single Server Queueing System



- Let's simulate the system from the scratch without any tool support

Overall Flow of Simulation Design

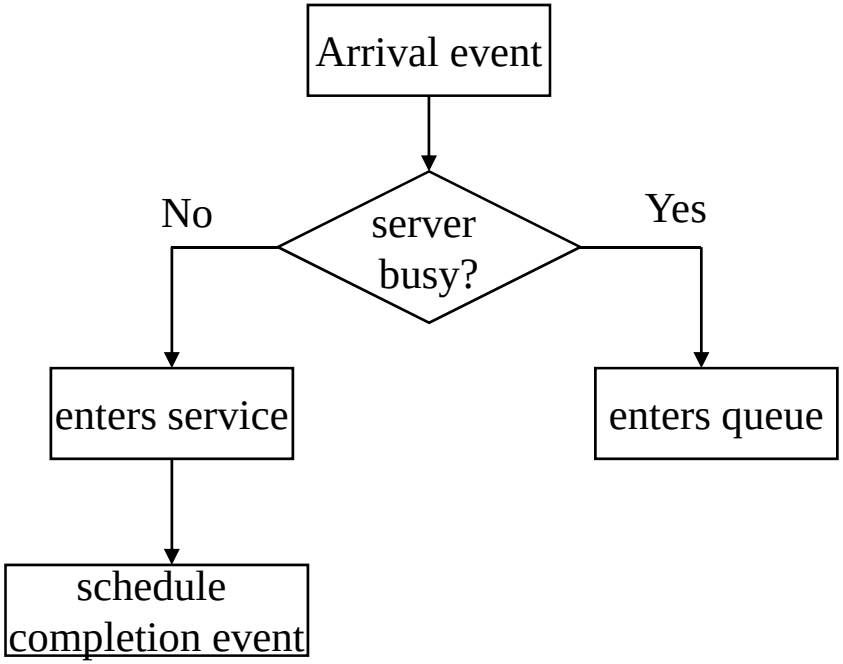
- Event modeling
  - Event type definition (What kinds of events happen in the system?)
  - Event handler definition (What is system's reaction for each event?)
- Implementation
  - Data structure for events (or jobs)
  - Data structure for simulating an object of the system
  - Data structure for collecting the necessary measurements
  - Data structure for schedule and dispatching events
- Data collection
  - average
  - timed average

What are the events in M/M/1?

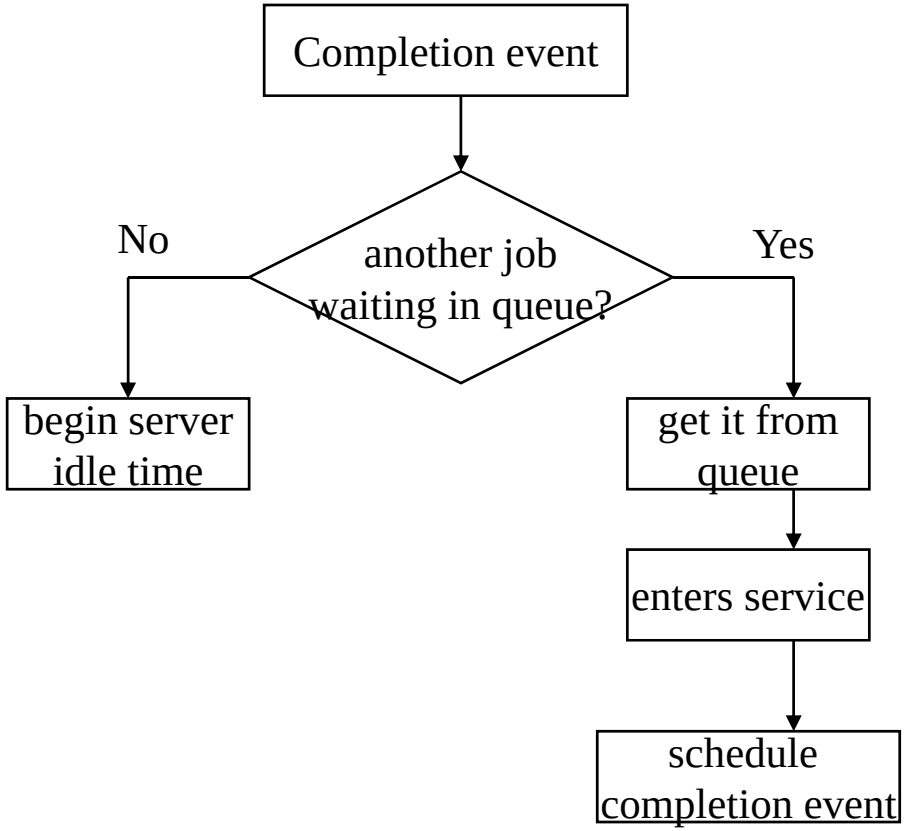
- Job arrival
- Service completion

What to do when an event occurs?

Job arrival event



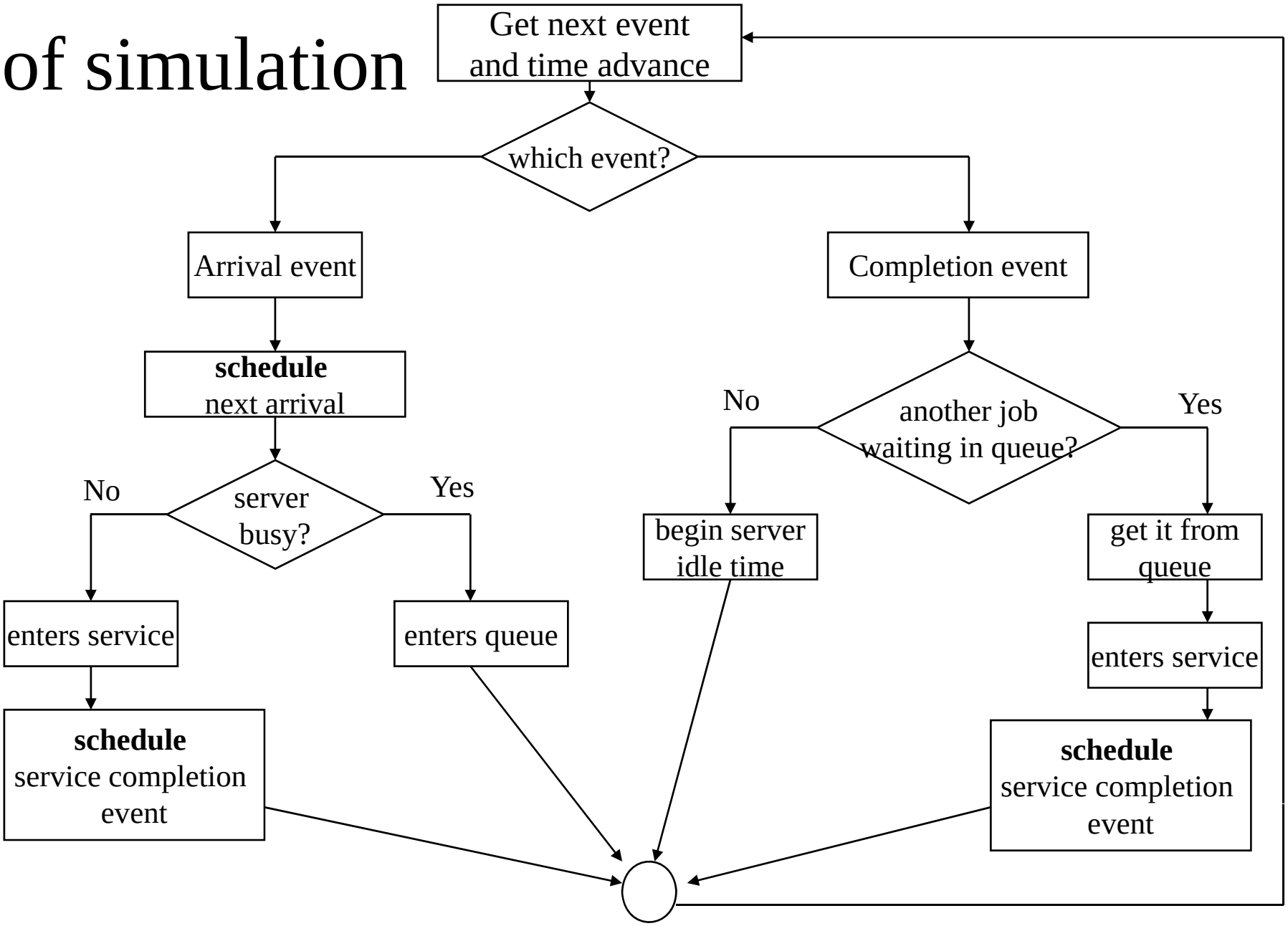
Service completion event



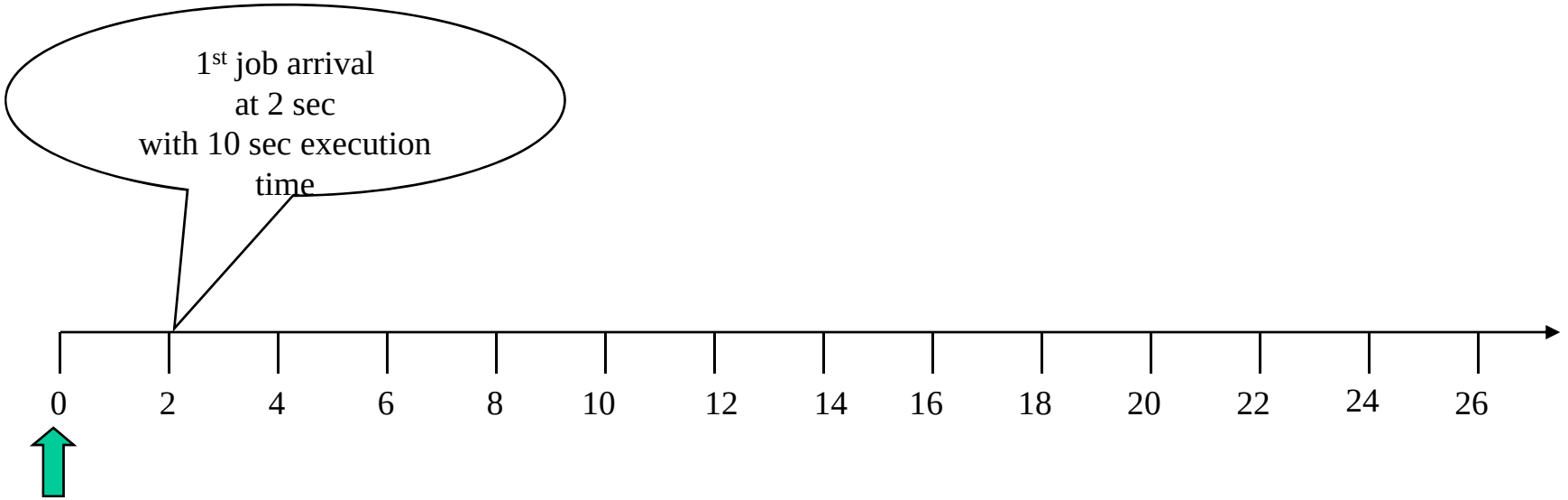
What data structure?

- Job
  - arrival time
  - execution time
- Server
  - busy/idle state variable
  - # of jobs served
- Queue
  - list of waiting jobs (FIFO with array?)

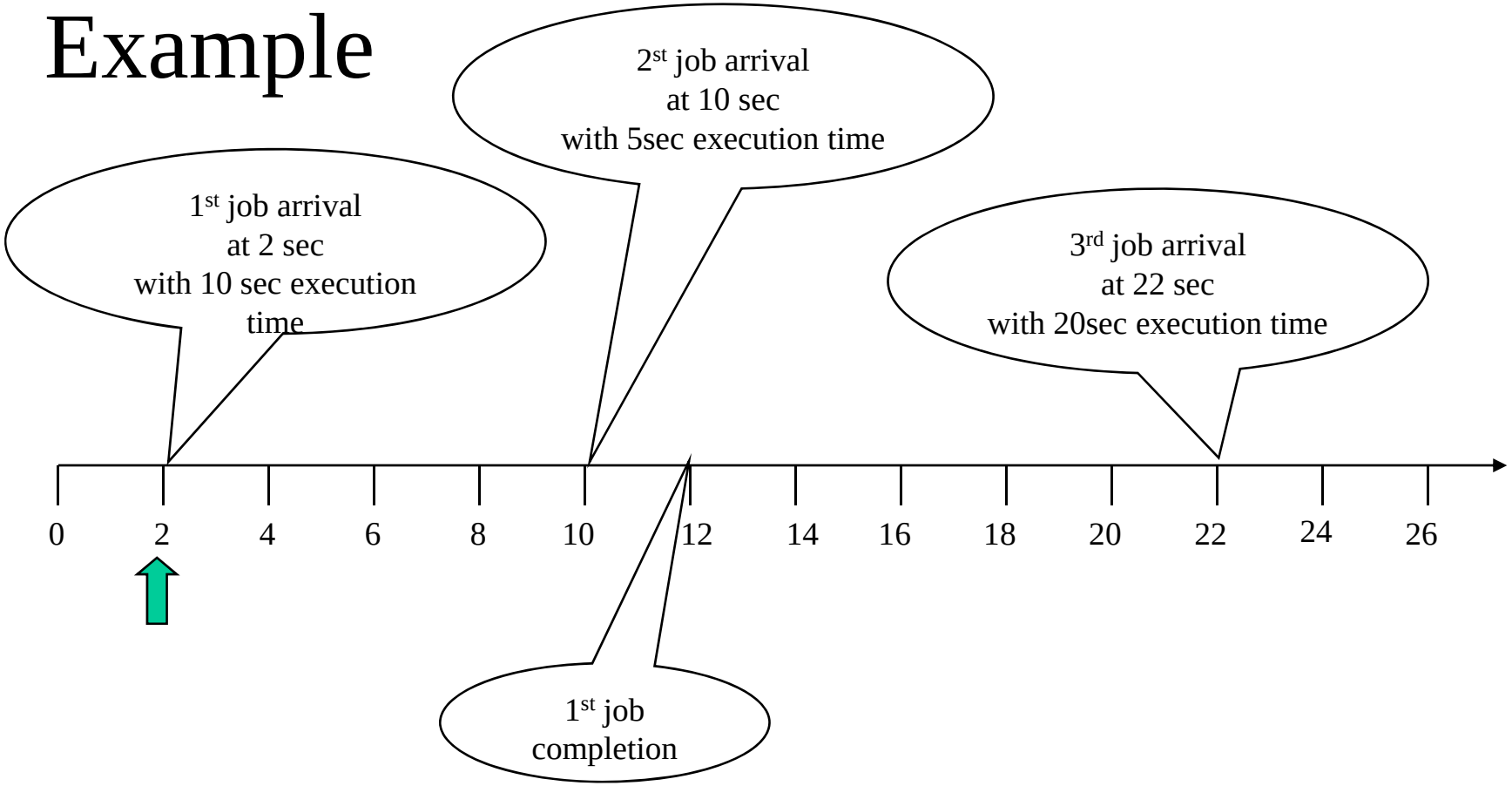
Overall flow of simulation



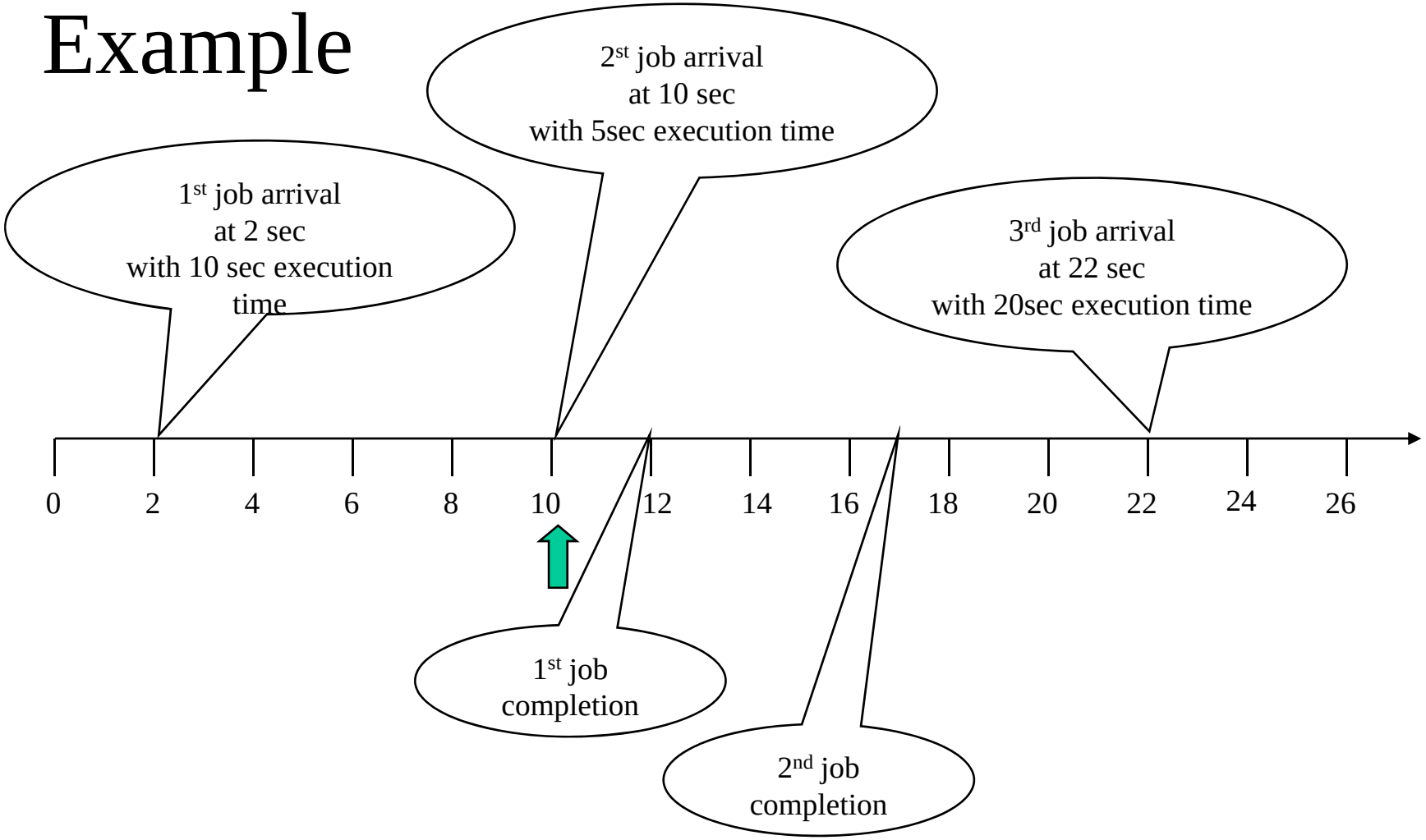
Example


1st job arrival
at 2 sec
with 10 sec execution
time
0
2
4
6
8
10
12
14
16
18
20
22
24
26

Example


2st job arrival
at 10 sec
with 5sec execution time
1st job arrival
at 2 sec
with 10 sec execution
time
3rd job arrival
at 22 sec
with 20sec execution time
0
2
4
6
8
10
12
14
16
18
20
22
24
26
1st job
completion

Example


2st job arrival
at 10 sec
with 5sec execution time
1st job arrival
at 2 sec
with 10 sec execution
time
3rd job arrival
at 22 sec
with 20sec execution time
0
2
4
6
8
10
12
14
16
18
20
22
24
26
1st job
completion
2nd job
completion

More data structure?

- Event
  - event type
  - occurrence time
- Event list
  - list of scheduled events in ascending order of time
  - Linked list

What generic functions?

- For event schedule
  - scheduleEvent(event, time)
  - event = getEvent();
- For random numbers
  - inter-arrival times following exponential distribution
  - job execution times following exponential distribution

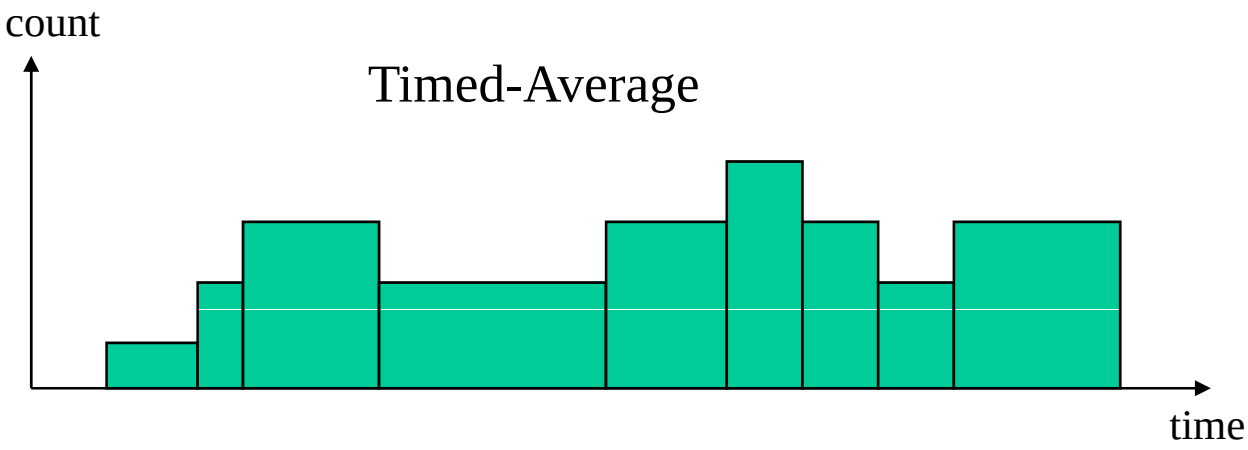
```
double ranf()  // return random number from uniform(0,1)
{
     return (double) rand() / (double) RAND_MAX;
}

double expntl(double x)
   { /* 'expntl' returns a psuedo-random variate from a negative     */
     /* exponential distribution with mean x.                        */
     return(-x*log(ranf()));
   }
```

Summary of what to do

- Generic (system independent)
  - scheduleEvent(event, time)
  - event = getEvent();
  - random number generators
- System-specific
  - Define events
  - Define the operation steps for each event
  - Determine the most appropriate data structure

What we want to see?

- Average delay of each customer
  - for each customer (job), record the arrival-time and departure-time
  - average “departure-time – arrival-time” for all jobs
- Average number of customers in the system as increasing the traffic intensity
- Average queue length



$$TimedAverage = \frac{\sum count \times interval}{totalTime}$$

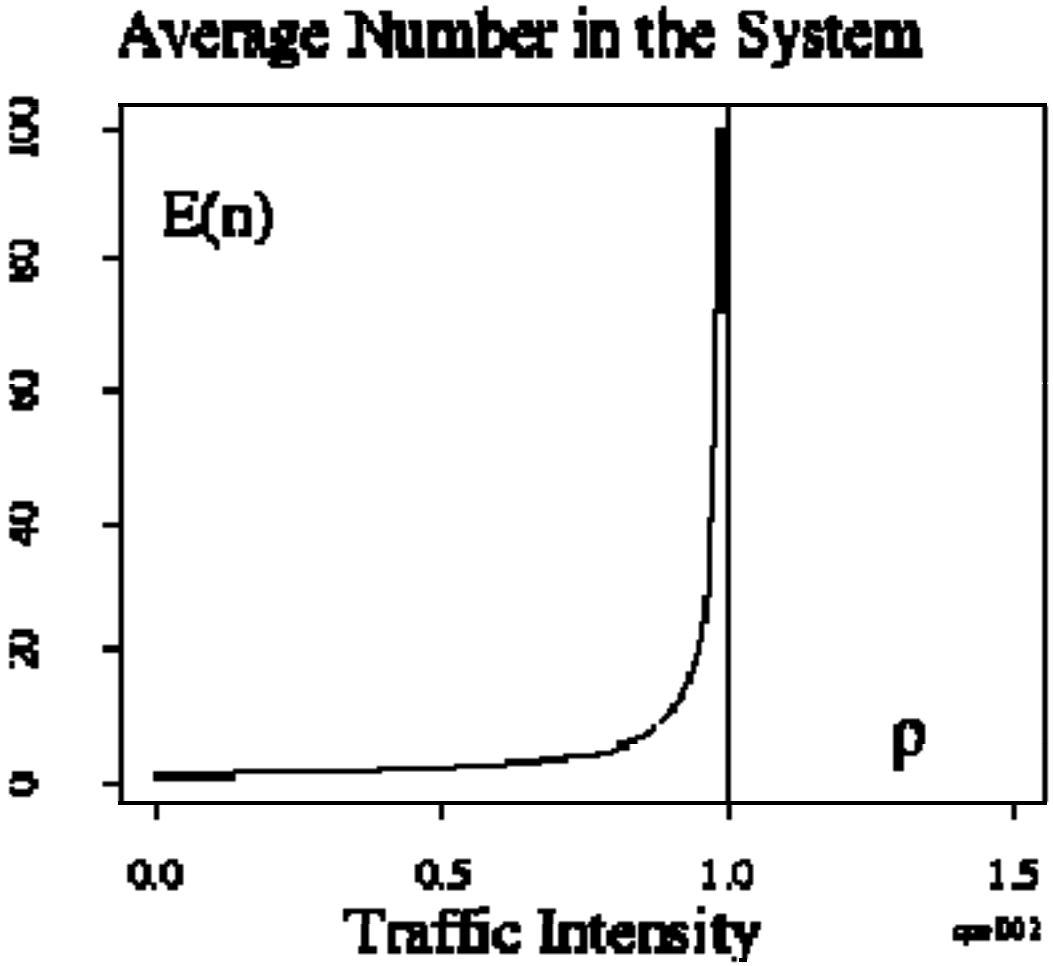
Simulation table

- Multiple simulation runs as changing the input parameters

Runs	Inputs				Inputs			
	X_{i1}	X_{i2}	…	…	Y_{i1}	Y_{i2}	…	…
1								
2								
…								
n								

- You can either use the outer for loop or shell script program

AvgNum as increasing ρ



Homework 1

- Draw the following three graphs by simulation
  - traffic load (0-0.95) vs. average # of jobs in the system
  - traffic load (0-0.95) vs. average queue length
  - traffic load (0-0.95) vs. average queueing delay for each job
- Draw the same three graphs by M/M/1 queueing analysis results
- When simulating the system with a traffic load 1.0 and 1.5,
  - Observe what happens in your simulation
  - Discuss how to handle such situation