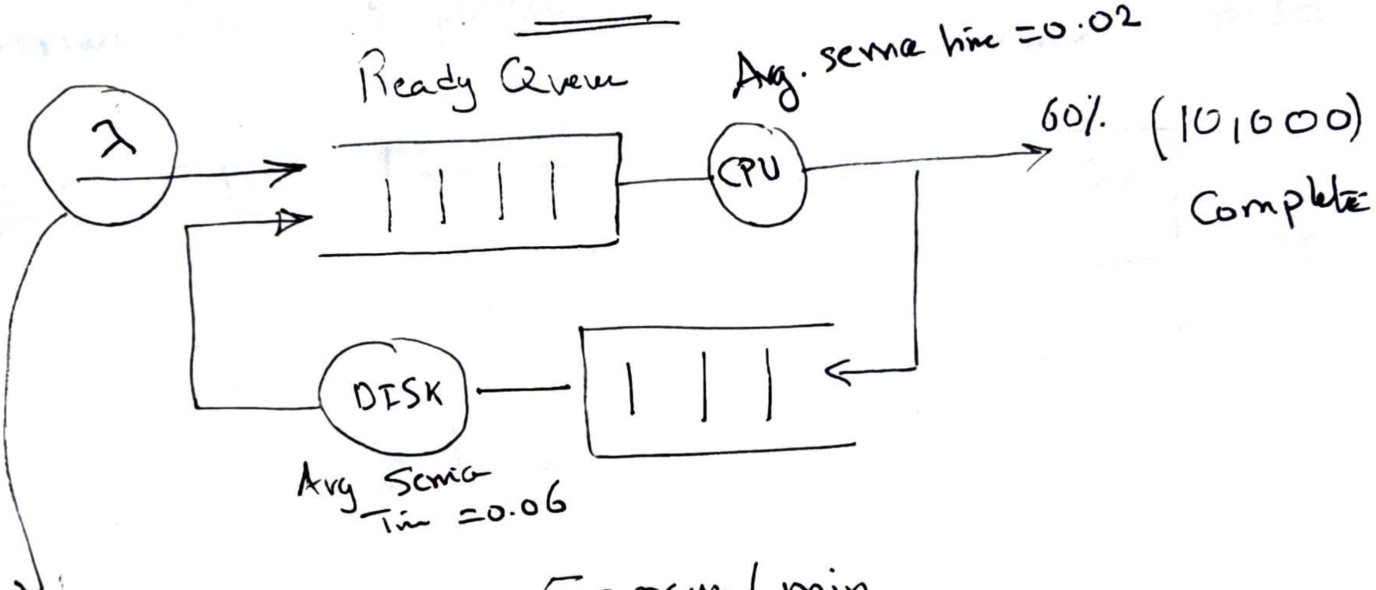
Problem 2 in HW#2.

n processes $i = 1$ — n

X_i : service time requested by process i



arrival rate : 5 process / min

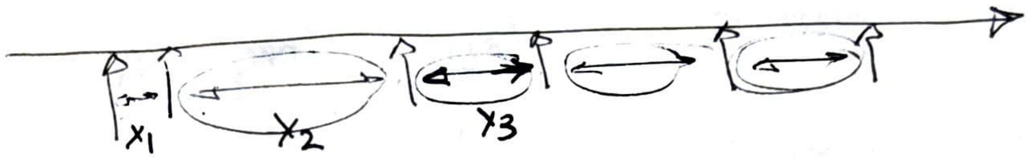
Poisson Dist

λ: rate parameter

Exp. Dist

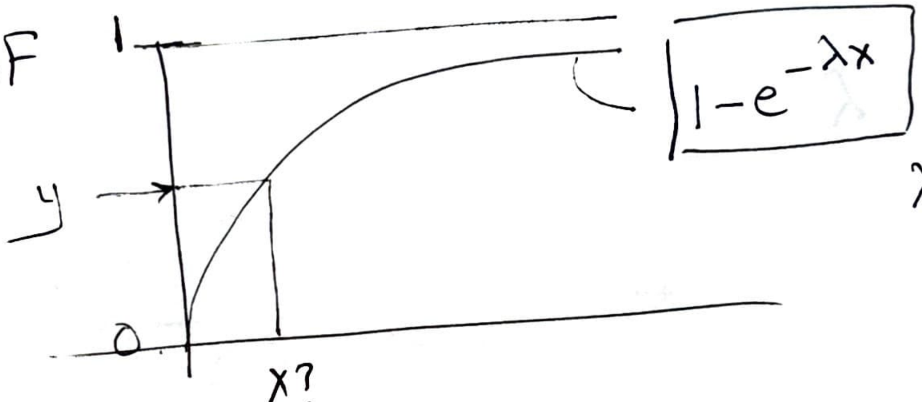
Service times for the process on the CPU or Disk

Memoryless property.



λ = arrival rate = 2 process/min $\Rightarrow$ avg interarrival time = 30 sec.

<u>Exp</u>: CDF



avg.
λ = rate

$$y = 1 - e^{-\lambda x}$$ $\longrightarrow$ Find x.

$$e^{-\lambda x} = 1 - y$$

$$-\lambda x = \ln(1-y)$$

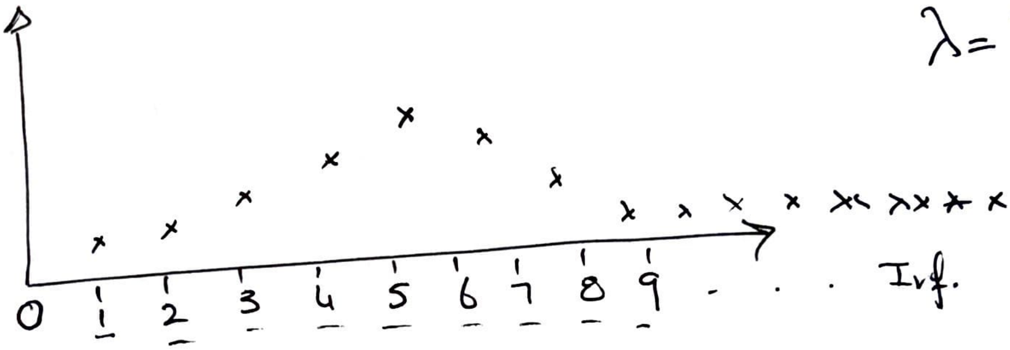
$$x = \frac{-1}{\lambda} \ln(1-y)$$

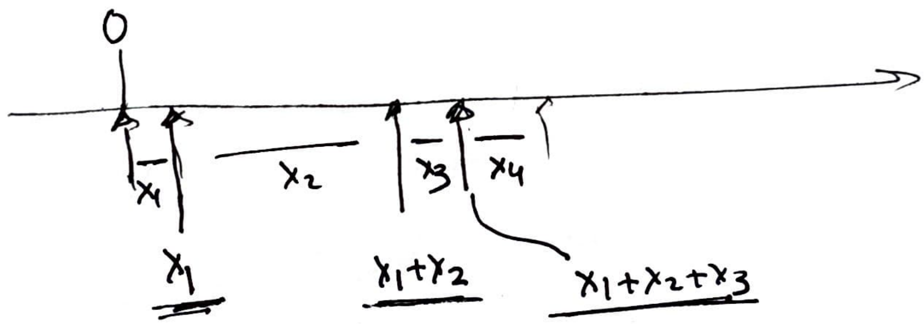
$\hookrightarrow$ uniform random number between 0 and 1

Poisson Dist.

gives the prob that a certain number of events happen if the average rate is known (λ).

$\lambda = 5$ process/min





For Service Times

CPU $[0.02]$ average. $\Rightarrow \dfrac{1}{0.02} \Rightarrow \boxed{50}$ rate process/sec

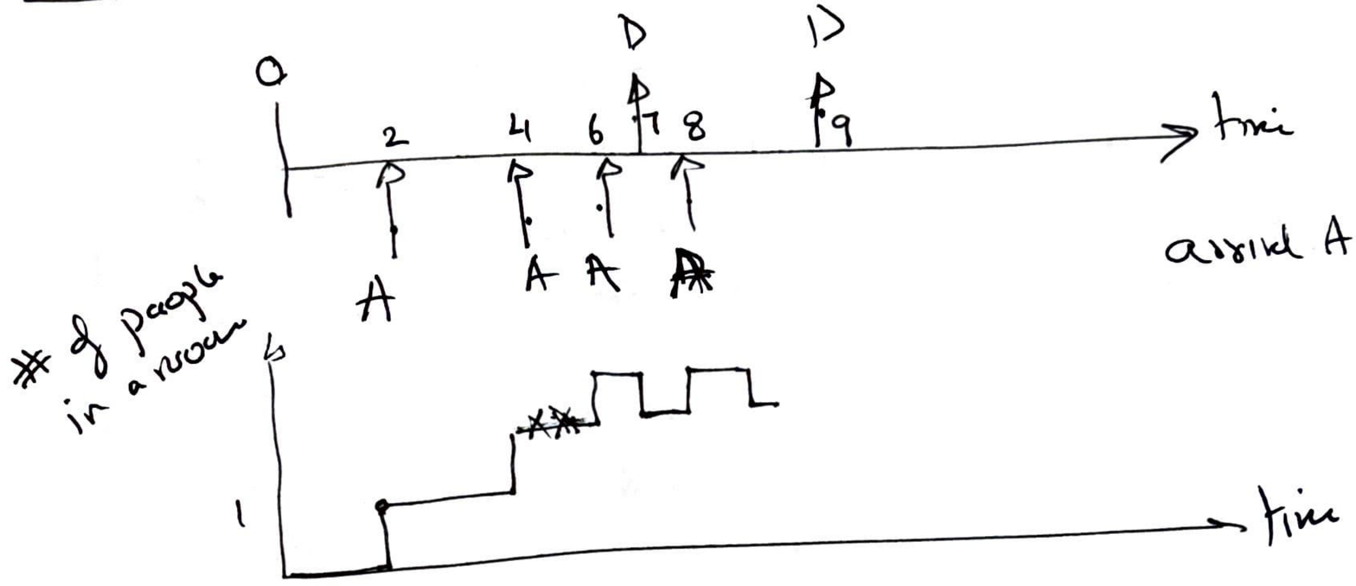
---

To generate Service times for the CPU.

$y \Leftarrow rand(0,1)$

$$\text{servicetime} = \frac{-1}{50}\left(\ln(1-\underset{0.7}{y})\right) = — \quad 0.024$$

Discrete-time event Simulation



1 Events

Things that happen (arrival, deperature,).

2 State:

bunch of variables / data structure that capture the system at a point in time.

State gets updated when an event happens.

3 Clock:

variable that keeps track of the current event.

hops between events.

Structure of Simulation

Initialization

State
CLOCK = 0
End-cond
Create first event (e.g. arrival of first process).

Running the Simulation Engine

while (! end-cond) {
~~End-cond~~
}

Statistics.

Event Queue

Data Structures (linked list or priority queue). Used by the discrete-time event simulation to track events.

Ordered in time

```
struct event {
        double time;
        int    type;
        int    processID;
        :      :
        event * next;
}
```