Chapter 4

- We’ll work with sim (nib)
- We’ll do some examples
- We’ll skip everything else

Simulation Package (nib)

- Package sim
- Implement event scheduling
- Derive classes to implement details
- See OneServer3.java for a simple sim-based simulator
Problem 31 ch. 4

- **State** = \{Q,s1,s1\}
  - Q.len = 0-5, Q is an actual queue
  - S1,S2 = 0|1 (busy or idle)
- **Events** = \{A,F1,F2\}
- **R_A:**
  - If s1==0, set s1=1 → create F1(t+[8+-2])
  - Else if s2==0, set s2=1 → create F2(t+[12+-4])
  - Else if Q.len<5 enqueue
  - Else with prob. 0.3 generate A(t+[60+-20])
  - Create A(t+4.5)

Problem 31

- **R_F1:**
  - If Q.len>0 dequeue customer; create F1(t+[8+-2])
  - Else set s1=0
- **R_F2:**
  - If Q.len==0 set s2=0
  - Else if(s1 == 0) dequeue, set s1=1, s2=0; create F1(t+[8+-2])
  - Else dequeue, set s2=1; create F2(t+[12+-4])
Problem 31

• Statistics:
  – (a) count total turned away/total time (T)
  – (b) count turned away that didn’t come back/T
  – (c) record arrival time and departure time of each customer/total customers served
  – (d) sum all finish times/total customers served
  – (e) sum Q.len*d_t/T. d_t time between events

Problem 31

• If you implements this, FEL blows up!
• Because returning customers generate next arrivals which generate returning customers the actual arrival rate grows exponentially!
• Try it with P31.java (comment out the code which fixes this…)
• We have to generate arrivals at a fixed rate and superimpose return events on this somehow
Problem 31

Let $K$ be arrivals/hr. Initially $K_0 = 60/4.5$. Assume a fraction $r$ of the time the queue is full. This means after 1 hour $r \cdot 0.3 \cdot K_0$ people return, some of which, say some number $0 < s < 1$, will return the next hour.

This means $K_1 = K_0 + s \cdot r \cdot 0.3 \cdot K_0$

$\Rightarrow$ Exponential growth!

Problem 31: fixed

- **State** = \{Q,s1,s2\}
  - Q.len = 0-5, Q is an actual queue
  - S1,S2 = 0|1 (busy or idle)
- **Events** = \{A,F1,F2\}
  - A has a field indicating if is return customer
- **R_A:**
  - If s1==0, set s1=1 → create F1(t+[8+-2])
  - Else if s2==0, set s2=1 → create F2(t+[12+-4])
  - Else if Q.len<5 enqueue
  - Else with prob. 0.3 generate A(t+[60+-20])
    - Mark this newly generated A as a return customer
  - Create A(t+4.5) **only** if customer was not a returning one
Problem 54 from ed.3

• A 1-server 1-queue system where jobs arrive every $X \pm 0.5X$ seconds and are processed in $80 \pm 80$ seconds at the server. Determine $X$ such that server utilization is 90%
• See P54.java