Problem 5.12 (a) Since
\[ P_0 = \frac{1/\lambda}{1/\lambda + 1/\mu} = \frac{\mu}{\lambda + \mu} \]

it follows that
\[ \lim_{t \to \infty} \frac{N(t)}{t} = \frac{\alpha_0 \mu}{\lambda + \mu} + \frac{\alpha_1 \lambda}{\lambda + \mu} . \]

(b) The expected total time spent in state 0 by \( t \) is
\[ \int_0^t P_{00}(s)ds = \frac{\mu}{\lambda + \mu} t + \frac{\lambda}{(\lambda + \mu)^2} \left( 1 - e^{-(\lambda+\mu)t} \right) . \]

Calling the above \( E[T(t)] \) we have
\[ E[N(t)] = \alpha_0 E[T(t)] + \alpha_1 (t - E[T(t)]) . \]

Problem 5.19 Let \( R_0 \) be the time required to return to state 0 (starting in state 0), once it has been left (i.e., starting from the moment that the CTMC first leaves state 0). What is the expected value, \( E[R_0] \)?

The times spent in state 0 and away from state 0 constitute an alternating renewal process. So this can be answered by an elementary application of renewal theory. For that purpose, let \( \nu_0 = \sum_{j \neq 0} Q_{0,j} = -Q_{0,0} \) and let \( \alpha_j \) be the stationary or steady-state probability of state \( j \). Then
\[ \alpha_0 = \frac{1/\nu_0}{E[R_0] + (1/\nu_0)} , \]
from which we see that
\[ E[R_0] = \frac{1 - \alpha_0}{\alpha_0 \nu_0} . \]

Let \( Y \) be the time of the first transition, and condition on that time, to get
\[ E[T] = te^{-\nu_0 t} + \int_0^t E[T|Y = s]\nu_0 e^{-\nu_0 s} ds \]
\[ = te^{-\nu_0 t} + \int_0^t (s + E[R_0] + E[T])\nu_0 e^{-\nu_0 s} ds . \]

We can thus solve for \( E[T] \), getting
\[ E[T] = \frac{te^{-\nu_0 t} + [(1 - \alpha_0)/(\alpha_0 \nu_0)](1 - e^{-\nu_0 t}) + \int_0^t s\nu_0 e^{-\nu_0 s} ds}{e^{-\nu_0 t}} . \]
It would be OK to stop there, but you could go on to evaluate the integral. That leads to some simplification. Using the fact that \( \nu^2 s e^{-\nu s} \) is the density of the Erlang \( E_2 \) (special form of gamma, with shape parameter 2) distribution, we have

\[
\int_0^t s \nu_0 e^{-\nu_0 s} \, ds = \frac{1 - e^{-\nu_0 t}}{\nu_0} - \frac{\nu_0 t e^{-\nu_0 t}}{\nu_0}
\]

or

\[
\int_0^t s \nu_0 e^{-\nu_0 s} \, ds = \frac{1 - e^{-\nu_0 t}}{\nu_0} - t e^{-\nu_0 t}.
\]

When we apply the algebra, we get

\[
E[T_0] = \frac{e^{\nu_0 t} - 1}{\alpha_0 \nu_0}.
\]

**Problem 5.22** The analysis on page 153-154 with

\[
\lambda_n = \begin{cases} \lambda & n \geq 0 \\ \mu_n & 1 \leq n \leq s \\ s \mu & n > s \end{cases}
\]

We need \( \lambda < s \mu \).

**Problem 5.24** Let \( X_i(t) \) denote the number of customers at server \( i, i = 1, 2 \), when there is unlimited waiting room. The, in steady state,

\[
P(n_i \text{ at server } i) = \prod_{i=1}^{2} \left( \frac{\lambda_i}{\mu_i} \right)^{n_i} \left( 1 - \frac{\lambda_i}{\mu_i} \right)^{1 - n_i}.
\]

Now the model under consideration is just a truncation of the above, which is time reversible by problem 5.23. The truncation is to the set

\[
A \triangleq \{(n, m) : n = 0, m \leq N + 1 \text{ or } m = 0, n \leq N + 1 \text{ or } n > 0, m > 0, n + m \leq N + 2\}.
\]

Hence, for \( (n, m) \in A \),

\[
P(n, m) = C \left( \frac{\lambda_1}{\mu_1} \right)^n \left( 1 - \frac{\lambda_1}{\mu_1} \right) \left( \frac{\lambda_2}{\mu_2} \right)^m \left( 1 - \frac{\lambda_2}{\mu_2} \right)^{n + m}.
\]

**Problem 5.25** In steady state it has the same probability structure as the arrival process. Hence if we include in the departure process those arrivals that do not enter, then it is a Poisson process.

**Problem 5.26 (a)** Follows from results of section 6.2 by writing \( n' = D_j n \) and so \( n = B_{j-1} n' \).
(b) A Poisson process by time reversibility. If \( D(0) = 0 \), it is a nonhomogeneous Poisson process.

**Problem 5.28** For \( \mathbf{n} = (n_1, \cdots, n_r) \), let

\[
P(\mathbf{n}) = C \prod_{i=1}^{r} \alpha_{n_i}^{n_i}
\]

If

\[
\mathbf{n}' = (n_1, \cdots, n_i - 1, \cdots, n_j + 1, \cdots, n_r)
\]

with \( n_i > 0 \) then

\[
P(\mathbf{n}) q(\mathbf{n}, \mathbf{n}') = P(\mathbf{n}') q(\mathbf{n}', \mathbf{n})
\]

\[
\iff \quad \alpha_i \frac{\mu_i}{r-1} = \alpha_j \frac{\mu_j}{r-1}
\]

\[
\iff \quad \alpha_i \mu_i = \alpha_j \mu_j
\]

So, setting \( \alpha_i = 1/\mu_i \), \( i = 1, \cdots, r \) and letting \( C \) be such that \( \sum_{\mathbf{n}} P(\mathbf{n}) = 1 \) the time reversibility equations are satisfied.

**Problem Extra**  Program = Calculations for the M/M/s/r=M Model following 1999 Improving paper

WhoWhen = Ward Whitt 11/17/03

Step1 = Model Parameters
s = number of servers, \( s = 100 \)
\( \lambda = \) arrival rate,\( \lambda = 100 \)
\( \mu = \) individual service rate, \( \mu = 1 \)
\( \alpha = \) individual abandonment rate (exponential abandonments), \( \alpha = 1 \)
r = number of extra waiting spaces, \( r = 100 \)

Step2 = Key Steady-state Probabilities
ProbNoWait = Probability of not having to wait before beginning service \( \text{ProbNoWait} = 0.48670120172085 \)
ProbAllBusy = Probability that all servers are busy upon arrival \( \text{ProbAllBusy} = 0.61329879827915 \)
ProbWaitandServed = 0.47343780147000
ProbLoss = Probability that an arrival is lost (blocked) at arrival \( \text{ProbLoss} = 4.71697062792618e-019 \)
ProbAban = Probability that a customer eventually abandons \( \text{ProbAban} = 0.03986099680915 \)

Step3 = Mean Values - Counting
MeanInQueue = 3.98609969091471
MeanBusyServers = 96.0139031906522
MeanNumberinSys = 99.99999999999993

Step4 = Second Moments and Variances - Counting
SecondMomInQueue = 61.3298792791486
VarInQueue = 35.44088916172650
SecondMomBusyServers = 9.254150183989135e+003
VarBusyServers = 32.78112950590003
SecondMomNumberinSys = 1.010000000000000e+004
VarNumberinSys = 1.000000000000000e+002

Step5 = Response-Time Moments
MeanRespTime = 0.99750428450446
CondMeanRespTimeServed = 1.03891653311596
SecMomRespTime = 2.94660549401905
VarRespTime = 1.951659129641430
CondSecMomRespTimeServed = 3.06993698123555
CondVarRespTimeServed = 1.98958941845386

Step6 = Waiting-Time Moments
Step6a = Waiting Times for Customers who are Served
MeanWaitServed = 0.03736528131360
CondMeanWaitServed = 0.03891653311596
Step 6b = Waiting Times for Customers who Abandon
MeanWaitAbandon = 0.00249571549554
ConditMeanWaitAban = 0.06261046374463
SecMomWaitAbandon = 2.695089209549688e-004
ConditSecMomWaitAbandon = 0.00676121879855
ConditVarWaitAban = 0.00284114862823

Step 6c = Waiting Time for All Customers, Served or Abandoning
MeanWaitTimeAll = 0.03986099680915
SecMomWaitTimeAll = 0.00499143099109
VarWaitTimeAll = 0.00340253192447

Step 7 = Conditional Waiting Time CCDF Values For Served Customers
\( t = 0.05000000000000 \) ProbOKWaitifServed = 0.70144791424012
\( t = 0.10000000000000 \) ProbOKWaitifServed = 0.97658087743471
\( t = 0.20000000000000 \) ProbOKWaitifServed = 0.99993119489496

Step 8 = Conditional Waiting Time CDF Values For Customers Who Abandon
\( t = 0.05000000000000 \) ProbOKWaitifAbandon = 0.50953673688750
\( t = 0.10000000000000 \) ProbOKWaitifAbandon = 0.79190403053181
\( t = 0.20000000000000 \) ProbOKWaitifAbandon = 0.97645578331304
\( t = 0.40000000000000 \) ProbOKWaitifAbandon = 0.99995439602867

Step 9 = Waiting Time CDF Values For All Customers
\( t = 0.05000000000000 \) ProbOKWait = 0.69379814341203
\( t = 0.10000000000000 \) ProbOKWait = 0.84522061993647
\( t = 0.20000000000000 \) ProbOKWait = 0.97657589105832
\( t = 0.40000000000000 \) ProbOKWait = 0.99993211971528

(a) MeanInQueue = 3.98609968091471, VarInQueue = 35.44088916172650
(b) arrival rate × ProbAban = 100 × 0.03986099680915 = 3.986
(c) ProbNoWait = 0.48670120172085
(d) ConditMeanWaitServed + service time = 0.03891653311596 + 1
(e) Conditional Waiting Time CDF Values For Served Customers at \( t = 0.1 \) = ProbOK-WaitifServed = 0.84743410401854
(f) Since the abandon rate \( \alpha \) equals the service rate \( \mu \), the model simplifies. But note that the simplification and following solution only works for the special case of \( \alpha = \mu \). First, in the view point of the number of customers in the system, the system is equivalent to \( M/M/s + r/0 \) since the departure rate from the system including the abandoned customers is \( n\mu \) even if \( n > s \) because of \( (n-s)\alpha = (n-s)\mu \) abandonment rate. However, we can anticipate that blocking is negligible with such a large waiting space. Thus the \( M/M/s + r/0 \) model should be essentially equivalent to a \( M/M/\infty \) model, for which the steady-state distribution is exactly Poisson, and approximately normal. Hence, the probability all servers are busy in the given \( M/M/s/r + M \) model is approximately equal to the probability that at least \( s \) servers are busy in the \( M/M/\infty \) model. Clearly, it is possible to compute the probability that at least \( s \) servers are busy in the \( M/M/\infty \) mode very quickly. (It might not be regarded as “simple,” however. When you read, “it is easy to see that . . . ,” in a paper, take that as an invitation to check very carefully.)
Now we describe how to proceed with the more special, but exact, representation in terms of the $M/M/s + r/0$ model: For $n < s + r = 200$

\[
P_n = \frac{\lambda_0 \cdots \lambda_{n-1}}{\mu_1 \cdots \mu_n} \left( 1 + \sum_{k=1}^{s+r} \frac{\lambda_0 \cdots \lambda_{k-1}}{\mu_1 \cdots \mu_k} \right)
\]

\[
= \frac{\lambda^n}{n! \left( \sum_{k=0}^{s+r} \frac{\lambda^k}{k!} \right)}
\]

\[
= \frac{e^{-\lambda} \lambda^n}{\left( \sum_{k=0}^{s+r} e^{-\lambda} \frac{\lambda^k}{k!} \right)}
\]

and we have (for Poisson random variable $N$ with parameter $\lambda = 100$)

\[
P(\text{All servers are busy}) = \sum_{n=0}^{s-1} P_n
\]

\[
= \frac{P(N \leq 99)}{P(N \leq 200)}
\]

\[
= 0.48670120172087 \text{ from MATLAB}
\]

\[\text{(g) MeanInQueue} \times \text{abandon rate per customer in queue}
\]

\[= 3.98609968091471 \times 1 = 3.98609968091471\]