1. Unexaminable (GSMPs not covered)
2. a)

i) GBEs:

lambda \* p0 = mu \* p1

(lambda + n \* mu) \* pn = (n+1) \* mu + p(n+1) + lambda \* p(n-1), 1 <= n <= c-1,

(lambda + c \* mu) \* pn = c \* mu \* p(n+1) + lambda \* p(n-1), c <= n < k.

c \* mu \* p(k) = lambda \* p(k-1), n = k

Can substitute given equations into LHS and RHS for each case and show this is true.

ii) We only consider k elements for our bounded queue (so P(n) = 0 for n > k) and so the sum of our k probabilities must equal 1 whereas for an unbounded queue we have the sum from 0 to infinity of pn must equal 1. If we define p0 as 1 - sum (1 to k) pn for bounded k and 1 - sum (1 to infinity) pn for an unbounded queue then p0 will differ for the bounded and unbounded queues.

b)

i) Area of trapezium = ½ (h + c) \* (b - a)

We require ½ (h+c) \* (b-a) = 1 => (h+c) \* (b-a) = 2

ii) ?? (Tried f(x) = ½ (h+c) \* (x-a) but this does not account for the new height of the RHS for x < b)

iii) Integrate f(x) from ii)

iv) For this case we get h = ⅖ by our normalising condition. We can scale U(0,1) by multiplying by ⅖ (as h = ⅖ at highest point then ⅖ U(a,b) will cover the entire trapezium). We then sample X from f(x) using Inverse Transform method as an inverse F^-1 exists, which we can use by sampling X from U1 and finding F^-1(U1 \* [(b-a) + a]) and find U2 (where U1, U2 ~ U(0,1)) and determine whether U2 \* h(x) (h(x) = ⅖ ) <= f(x). If so then we accept X.

P(success) = area of trapezium / area of rectangle

= 1 / (4 \* ⅖) = 0.625

E(trials before success) = (1 - 0.625) / 0.625 = 0.6

and number of expected samples = 2 (0.6 + 1) = 3.2 U(0,1) samples expected.

3a) P(k) = G \* rho ^ k.

M = 1, N = 1.

We find G = g(n, m). We take v1 = 1. so x1 = v1/mu1 = 1/mu.

|  |  |  |
| --- | --- | --- |
| M/N | 0 | 1 |
| 1 | 1 | 1/mu |

Therefore G = 1/mu.

Mean queue length = sum of n = 0 to infinity of n \* pn = 1 \* G \* rho^1 = rho/mu.

b) i) mu > gamma as if gamma >= mu we have lambda 1 = gamma + ½ lambda 4 > mu which means U >= 1 which results in an unstable system.

ii) gamma = (gamma, 0, 0, 0)

Q\* = [ [0 ½ ½ 0]

[0 0 1 0]

[0 0 0 1]

[½ 0 0 0] ]

We get from traffic equations that lambda1 = gamma + ½ lambda4 (eq 1), lambda2 = ½ lambda1 (eq 2),

lambda3 = ½ lambda1 + lambda2 (eq 3), lambda4 = lambda3 (eq 4).

By substituting (4) into (3) and (3) into (1) we get lambda1 = 2 \* gamma, lambda2 = gamma, lambda4 = lambda3 = 2 \* gamma. Then s1 = s2 = 1/(4 \* gamma) and s3 = s4 = 1/(8 \* gamma)

v1 = lambda1/gamma = 2, v2 = lambda2/gamma = 1, v3 = lambda3/gamma = 2, v4 = lambda4/gamma = 2.

R = sum of i = 1 to n (vi \* wi) = 5/4 \* gamma seconds [where we take each service time as the waiting time].

ii) Then 2(1/(ɑ𝛾 - 2𝛾)) + 1/(ɑ𝛾 - 𝛾) + 2(1/(2ɑ𝛾 - 2𝛾)) + 2(1/(2ɑ𝛾 - 2𝛾)) < 1/𝛾

After a bit of algebra the required answer pops out