Robert Johns
October 2, 2013

CSCI 426: Simulation

Homework 3

3.1.5 (a) Verify that the mean service time in Example 3.1.4 is 1.5

Solution: Verified by modifying `ssq2.c` as instructed in Example 3.1.4. The additional code is a modified version of `GetService(void)` and a new function `Geometric(double p)`, which are written below:

```
long Geometric(double p) {
        return ((long) (log(1.0 - Random()) / log(p)));
}

double GetService(void) {
        long k;
        double sum = 0.0;
        long tasks = 1 + Geometric(0.9);

        for (k = 0; k < tasks; k++) {
                sum += Uniform(0.1, 0.2);
        }
        return(sum);
}
```

The additional code yielded a result ing $\bar{s}$ of 1.49, which is negligibly less than the given result of 1.50.

(b) Verify that the steady-state statistics in Example 3.1.4 seem to be correct.

Solution: Verified by running the program using the same code as above. The average wait, average delay, average number in the node and average number in the queue all differed slightly, though negligibly from the given results in the book. The statistics follow:

statistic	exercise value	example value
average interarrival time	2.00	2.00
average wai*t*	6.02	5.77
average delay	4.53	4.27
average service time	1.49	1.50
average in the node	3.02	2.89
average in the queue	2.27	2.14
utilization	0.75	0.75

(c) Note that the arrival rate, service rate, and utilization are the same as those in Example 3.1.3. Explain (or conjecture) why this is so.

Solution: We did not change `GetArrival(void)`, so the arrival rate should be the same. The service rate will be the same because the parameter chosen (0.9) for the geometric random variable will create a function similar to a uniform random variable, which is how the service time is calculated in Example 3.1.3. Utilization depends on arrival time and service rate, so it will be unchanged as the other two are unchanged.

3.2.1 (a) Construct the $a = 16807$ version of the table in Example 3.2.6.

Solution: We have $(a, m) = (16807, 2^{31} - 1)$. Our table will be:

s	$\lfloor m/s \rfloor$	j	$a^j \mod m$
1024	2097151	2085659	8208
512	4194303	4184337	374844
256	8388607	8335476	36563
128	16777215	16776028	188756

(b) What is the time complexity of the algorithm you used?

Solution: A constant-time set of operations is used for each n, so our algorithm is $O(n)$.

3.3.6 (a) Relative to Example 3.3.5, construct a figure or table illustrating how $\bar{x}$ (utilization) depends on M.

Solution:

M	$\bar{x}$
20	0.29
25	0.37
30	0.44
35	0.51
40	0.58
45	0.66
50	0.72
55	0.79
60	0.86
65	0.91
70	0.96
75	0.98
80	1.00
85	1.00
90	1.00
95	1.00
100	1.00

(b) If you extrapolate linearly from small values of M, at what value of M will saturation $\bar{x} = 1$ occur?

Solution: Examining the following table, we see that saturation occurs at $M = 80$.

M	$\bar{x}$
20	0.29
21	0.31
22	0.32
23	0.34
24	0.35
25	0.37
26	0.38
27	0.40
28	0.41
29	0.43
30	0.44
31	0.45
32	0.47
33	0.48

M	$\bar{x}$
34	0.50
35	0.51
36	0.53
37	0.54
38	0.56
39	0.57
40	0.58
41	0.60
42	0.61
43	0.63
44	0.64
45	0.66
46	0.67
47	0.68

M	$\bar{x}$
48	0.70
49	0.71
50	0.72
51	0.74
52	0.75
53	0.77
54	0.78
55	0.79
56	0.80
57	0.82
58	0.83
59	0.84
60	0.86
61	0.87

M	$\bar{x}$
62	0.88
63	0.89
64	0.90
65	0.91
66	0.92
67	0.93
68	0.94
69	0.95
70	0.96
71	0.96
72	0.97
73	0.98
74	0.98
75	0.98

M	$\bar{x}$
76	0.99
77	0.99
78	0.99
79	0.99
80	1.00
81	1.00
82	1.00
83	1.00
84	1.00
85	1.00
86	1.00
87	1.00
88	1.00
89	1.00

M	$\bar{x}$
90	1.00
91	1.00
92	1.00
93	1.00
94	1.00
95	1.00
96	1.00
97	1.00
98	1.00
99	1.00
100	1.00

(c) Can you provide an empirical argument or equation to justify this value?

Solution: There are several empirical arguments that justify 80 as the saturation value. First, a cursory examination of Figure 3.3.9 shows that the line $M - \bar{l}$ levels off around 80, and so there are a constant number of operational machines at any time for any $M \geq 80$. This is because, as we increase the number of machines without increasing the number of servers, more machines will fail and thus the server will be busier and busier until every machine is delayed. An equation for the fuction $f : M \to M - \bar{l}$ given by $M - \bar{l} = 67$ gives the maximum number of operational machines for any $M \geq 80$. Since this function is a flat line, the number of operational machines does not increase with M and so saturation will be 1 after 80.