

學號：N26111871 姓名：廖威任

繼模擬 M/M/1 之後本次作業加入 utilization、 average number of customers in queue、 average number of customers in system、 average time waiting in queue、 average time in system， 且分別比較 math 及 simulation 的結果。

```
49 #利用數學公式計算出utilization
50 rho = lam / mu
51 print("Server utilization(math):", rho)
52
53 #利用模擬的數據計算出utilization
54 busy = 0
55 for i in range(total_service):
56     busy += end_service[i] - start_service[i]
57 utilization = busy / t
58 print("Server utilization(simulation):", utilization)
```

1. 利用公式： $\rho = \lambda / \mu$ 來計算出 utilization。
2. 利用每位顧客開始服務的時間和結束服務的時間計算出 utilization。

```
60 #利用pasta算出路人的觀察時間
61 pasta = []
62 pasta.append((-1 / 10) * math.log(random.random()) * t)
63 i = 1
64 while(pasta[len(pasta) - 1] < t):
65     pasta.append(pasta[i - 1] + (-1 / 10) * math.log(random.random()) * t)
66     i += 1
```

路人觀察店內狀況的時間為 $\text{Exp}(10)$ 。

```

68 #利用數學公式計算出在queue中的平均顧客數
69 Nq = rho ** 2 / (1 - rho)
70 print("Average number of customers in queue(math):", Nq)
71
72 #利用模擬的數據計算出在queue中的平均顧客數
73 queue = []
74 for i in range(len(pasta)):
75     queue.append(0)
76     for j in range(total_service):
77         if arrival[j] < pasta[i] and start_service[j] > pasta[i]:
78             queue[i] += 1
79 avg_num_q = sum(queue) / len(queue)
80 print("Average number of customers in queue(simulation):", avg_num_q)

```

1. 利用公式： $Nq = \rho^2 / (1 - \rho)$ 計算出在 queue 中的平均顧客數。
2. 若顧客的抵達時間小於 pasta 且開始服務時間大於 pasta 則表示顧客在 queue 中。

```

82 #利用數學公式計算出在系統中的平均顧客數
83 Ns = rho / (1 - rho)
84 print("Average number of customers in system(math):", Ns)
85
86 #利用模擬的數據計算出在系統中的平均顧客數
87 system = []
88 for i in range(len(pasta)): # 每10個時間單位觀察一次系統中的顧客
89     system.append(0)
90     for j in range(total_service):
91         if arrival[j] < pasta[i] and end_service[j] > pasta[i]:
92             system[i] += 1
93 avg_num_sys = sum(system) / len(system)
94 print("Average number of customers in system(simulation):", avg_num_sys)

```

1. 利用公式： $Ns = \rho / (1 - \rho)$ 計算出在系統中的平均顧客數。
2. 若顧客的抵達時間小於 pasta 且結束服務時間大於 pasta 則表示顧客在系統中。

```

96  #利用數學公式計算出Average waiting time
97  Wq = rho / (mu - lam)
98  Wq *= t
99  print("Average waiting time(math):", Wq)
100
101  #利用模擬的數據計算出Average waiting time
102  waiting = [0] * total_service
103  for i in range(total_service): #計算出每位顧客的等待時間
104      waiting[i] = start_service[i] - arrival[i]
105  avg_waiting = sum(waiting) / total_service #計算出平均的等待時間
106  print("Average waiting time(simulation):", avg_waiting)

```

1. 利用公式： $Wq = \rho / (\mu - \lambda)$ 計算出顧客的平均等待時間。
2. 每位顧客的等待時間為開始服務時間減抵達時間，再計算出顧客的平均等待時間。

```

108  #利用數學公式計算出顧客在系統中的平均時間
109  W = 1 / (mu - lam)
110  W *= t
111  print("Average time in the system(math):", W)
112
113  #利用模擬的數據計算出顧客在系統中的平均時間
114  sys_time = [0] * total_service
115  for i in range(total_service): #計算出每位顧客在系統中的時間
116      sys_time[i] = end_service[i] - arrival[i]
117  avg_sys_time = sum(sys_time) / total_service #計算出在系統中的平均時間
118  print("Average time in the system(simulation):", avg_sys_time)

```

1. 利用公式： $W = 1 / (\mu - \lambda)$ 計算出顧客在系統中的平均時間。
2. 每位顧客在系統中的平均時間為結束服務時間減抵達時間，再計算出顧客在系統中的平均時間。

```

請輸入lambda:800
請輸入mu:1000
請輸入t:1000
Server utilization(math): 0.8
Server utilization(simulation): 0.7923855621298365
Average number of customers in queue(math): 3.2000000000000015
Average number of customers in queue(simulation): 2.9411764705882355
Average number of customers in system(math): 4.000000000000001
Average number of customers in system(simulation): 3.823529411764706
Average waiting time(math): 4.0
Average waiting time(simulation): 4.0020763372296715
Average time in the system(math): 5.0
Average time in the system(simulation): 4.992558289891957

```

此圖為其中一次的模擬結果。

在固定 $\lambda=800$ 、 $\mu=1000$ 、 $t=1000$ 的情況下跑 10 次模擬，並算出平均及誤差後結果如下：

	Math	Simulation	誤差
Utilization	0.8	0.8	0%
Average # of customers in queue	3.2	3.49	9.06%
Average # of customers in system	4	4.285	7.13%
Average waiting time	4	4.404	10.1%
Average time in the system	5	5.41	8.2%