



SAVEETHA ENGINEERING COLLEGE

AUTONOMOUS

Affiliated to Anna University | Approved by AICTE



LABORATORY RECORD NOTEBOOK

Name of the Student : PABBARTHI..CHETAN..SATHISH..KUMAR.....

Register Number : 212220230033.....

Department : ARTIFICIAL INTELLIGENCE AND DATA SCIENCE.....

Semester : IV [5U5-1].....

Subject : PROBABILITY AND QUEUING MODELS.....



SAVEETHA ENGINEERING COLLEGE

Affiliated to Anna University | Approved by AICTE

AUTONOMOUS

NAAC
A
GRADE

Accredited by
NBA
NATIONAL BOARD OF ACCREDITATION
5 US Programs

Department ofARTIFICIAL INTELLIGENCE...AND..DATA SCIENCE.....

LABORATORY RECORD NOTE BOOK

20²²- 20²³

It is to certify that this is a bonafide record of work done by

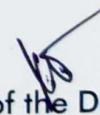
Mr. / Ms. PABBARTHI CHETAN SATHISH KUMAR. Register Number 212220230033...

of I / II / III / IV year B.E / B.Tech. / M.E / M.B.A. Department

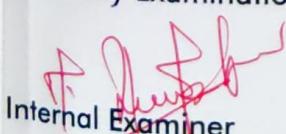
of ARTIFICIAL INTELLIGENCE & DATA SCIENCE. in the PROBABILITY & QUEUING MODELS

Laboratory in the IV [5US-1] Semester.


Staff in Charge


Head of the Department

University Examination held on 8/12/22


Internal Examiner


External Examiner



SAVEETHA ENGINEERING COLLEGE

AUTONOMOUS



Affiliated to Anna University | Approved by AICTE

EX.NO	DATE	INDEX	PAGE NO	INCHARGE SIGNATURE
1	30.08.22	Mean and variance of a discrete distribution	1	
2	13.09.22	Fitting Poisson distribution	3	
3	27.09.22	Correlation and regression for data analysis	10	
4	11.10.22	Single server with infinite capacity (M/M/1):(∞/FIFO)	19	
5	28.10.22	Multiple server with infinite capacity - (M/M/c):(∞/FIFO)	12	
6	15.11.22	Series Queues with infinite capacity - Open Jackson Network	18	

EXP NO.1

DATE:30/08/22

Mean and variance of a discrete distribution

Aim :

To find mean and variance of arrival of objects from the feeder using probability distribution

Software required :

Python and Visual components tool

Theory:

The expectation or the mean of a discrete random variable is a weighted average of all possible values of the random variable. The weights are the probabilities associated with the corresponding values. It is calculated as,

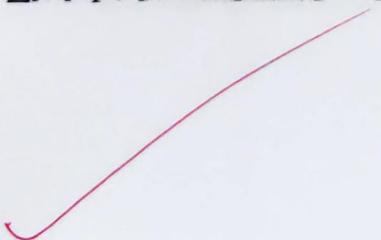
$$E(X) = \mu = \sum_i x_i p_i \quad i = 1, 2, \dots, n$$

$$E(X) = x_1 p_1 + x_2 p_2 + \dots + x_n p_n.$$

The variance of a random variable shows the variability or the scatterings of the random variables. It shows the distance of a random variable from its mean. It is calculated as

$$\sigma_x^2 = \text{Var}(X) = \sum_i (x_i - \mu)^2 p(x_i) = E(X - \mu)^2 \text{ or, } \text{Var}(X) = E(X^2) - [E(X)]^2.$$

$$E(X^2) = \sum_i x_i^2 p(x_i), \text{ and } [E(X)]^2 = [\sum_i x_i p(x_i)]^2 = \mu^2.$$



Procedure :

1. Construct frequency distribution for the data
2. Find the probability distribution from frequency distribution.
3. Calculate mean using

$$E(X) = \mu = \sum_i x_i p_i \quad i = 1, 2, \dots, n$$

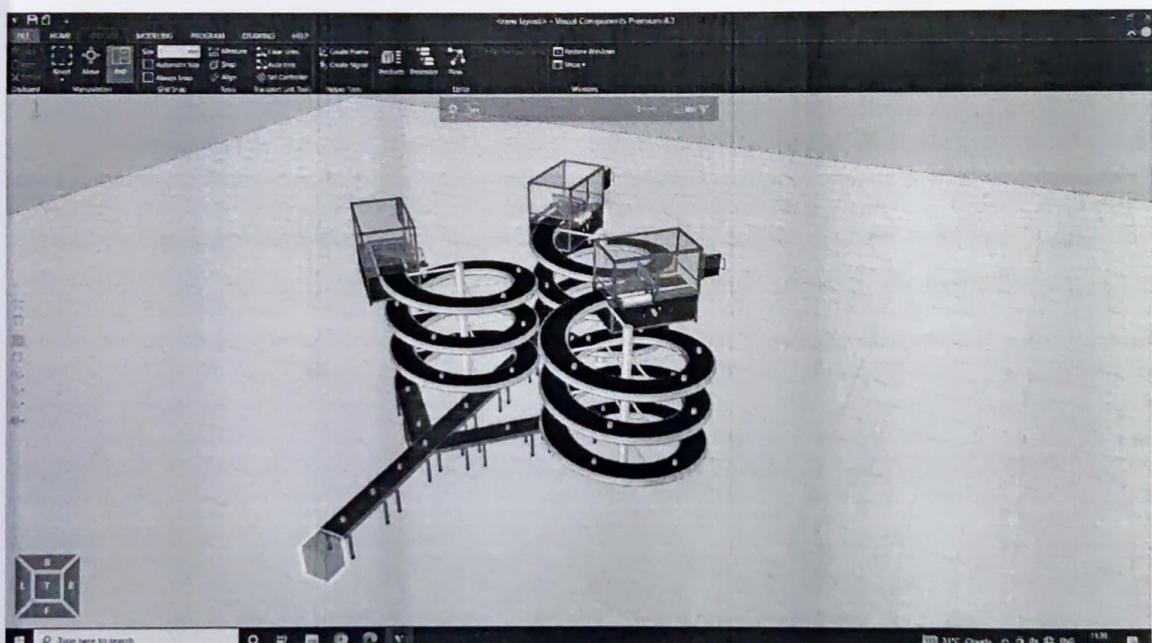
4. Find

$$E(X^2) = \sum_i x_i^2 p(x_i)$$

5. Calculate variance using

$$\text{Var}(X) = E(X^2) - [E(X)]^2$$

Experiment :



Program :

```
import numpy as np
L=[int(i) for i in input().split()]
N=len(L); M=max(L)
x=list();f=list()
for i in range (M+1):
    c = 0
    for j in range(N):
        if L[j]==i:
            c=c+1
    f.append(c)
    x.append(i)
sf=np.sum(f)
p=list()
for i in range(M+1):
    p.append(f[i]/sf)
mean=np.inner(x,p)
EX2=np.inner(np.square(x),p)
var=EX2-mean**2
SD=np.sqrt(var)
print("The Mean arrival rate is %.3f "%mean)
print("The Variance of arrival from feeder is %.3f "%var)
print("The Standard deviation of arrival from feeder is
%.3F "%SD)
```

Output :

```
8 8 7 6 4 8 9 6 5 2
The Mean arrival rate is 6.300
The Variance of arrival from feeder is 4.210
The Standard deviation of arrival from feeder is 2.052
```

Result :

The mean and variance of the arrival rate of feeder from
feeder is calculated.

EXP NO.2

DATE:13/09/22

Fitting Poisson distribution

Aim :

To fit poisson distribution for the arrival of objects per minute from the feeder

Software required :

Python and Visual component tool

Theory:

The Poisson distribution is the discrete probability distribution of the number of events occurring in a given time period, given the average number of times the event occurs over that time period.

If λ is mean, then the probability mass function of Poisson distribution is

$$P(X = x) = e^{-\lambda} \frac{\lambda^x}{x!}, \quad x = 0,1,2 \dots$$

Conditions for Poisson Distribution:

1. An event can occur any number of times during a time period.
2. Events occur independently.
3. The rate of occurrence is constant.
4. The probability of an event occurring is proportional to the length of the time period.

Procedure :

1. Compute mean $= \frac{\sum fx}{N}$, $N = \sum f$.

2. Calculate the expected frequencies from the probability mass function

$$P(X = x) = e^{-\lambda} \frac{\lambda^x}{x!}, \quad x = 0,1,2,\dots$$

3. Calculate the expected frequencies

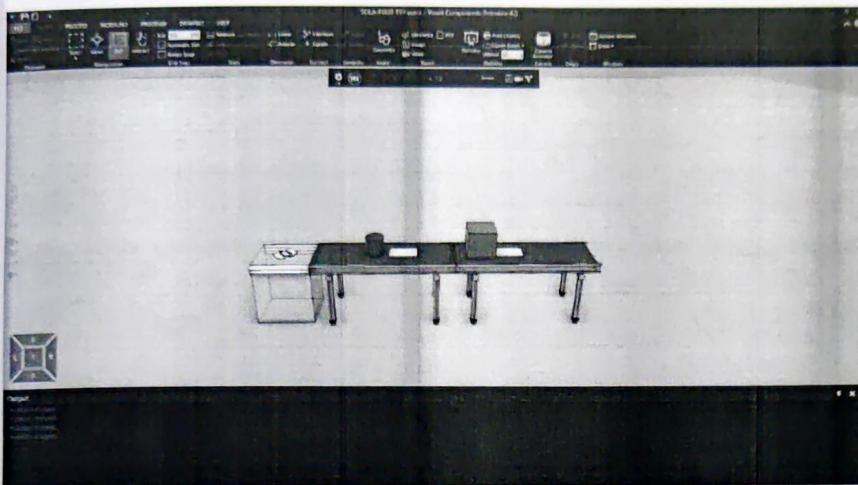
$$N \times P(X = x), \quad x = 0,1,2, \dots n$$

4. Calculate $\chi^2 = \sum \frac{(O-E)^2}{E}$, where O and E are observed and expected frequencies

5. Test χ^2 at 1% level of significance and write the conclusion



Experiment :



Program :

```
import numpy as np
import math
import scipy.stats
L=[int(i) for i in input().split()]
N=len(L); M=max(L)
X=list();f=list()
for i in range (M+1):
    c = 0
    for j in range(N):
        if L[j]==i:
            c=c+1
    f.append(c)
    X.append(i)
sf=np.sum(f)
p=list()
for i in range(M+1):
    p.append(f[i]/sf)
mean=np.inner(X,p)
p=list();E=list();xi=list()
print("X P(X=x) Obs.Fr Exp.Fr xi")
print("-----")
for x in range(M+1):
    p.append(math.exp(-mean)*mean**x/math.factorial(x))
    E.append(p[x]*sf)
    xi.append((f[x]-E[x])**2/E[x])
    print("%2.2f %2.3f %4.2f %3.2f %3.2f"%(x,p[x],f[x],E[x],xi[x]))
print("-----")
cal_chi2_sq=np.sum(xi)
print("Calculated value of Chi square is %4.2f"%cal_chi2_sq)
table_chi2=scipy.stats.chi2.ppf(1-.01,df=M)
print("Table value of chi square at 1 level is %4.2f"%table_chi2)
if cal_chi2_sq<table_chi2:
    print("The given data can be fitted in poisson Distribution at 1% LOS")
else:
    print("The given data cannot be fitted in Poisson Distribution at 1% LOS")
```

Output :

X	P(X=x)	Obs.Fr	Exp.Fr	xi
0.00	0.01	0.00	0.05	0.05
1.00	0.03	1.00	0.24	2.35
2.00	0.08	1.00	0.62	0.23
3.00	0.13	1.00	1.07	0.00
4.00	0.17	0.00	1.37	1.37
5.00	0.18	1.00	1.40	0.12
6.00	0.15	1.00	1.20	0.03
7.00	0.11	1.00	0.88	0.02
8.00	0.07	1.00	0.56	0.34
9.00	0.04	1.00	0.32	1.45

Calculated value of Chi square is 5.94

Table value of Chi square at 1 level is 21.67

The given data can be fitted in poisson distribution at 1% LOS

Result :

fitting poisson distribution for the arrival of objects per minute from the feeder is calculated.

EXP NO.3

DATE:27/09/22

Correlation and regression for data analysis

Aim :

To analyse given data using coefficient of correlation and regression

X	25	28	35	32	31	36	29	38	34	32
Y	43	46	49	41	36	32	31	30	33	39

line

Software required :

Python

Theory:

Correlation describes the strength of an association between two variables, and is completely symmetrical, the correlation between A and B is the same as the correlation between B and A. However, if the two variables are related it means that when one changes by a certain amount the other changes on an average by a certain amount.

If y represents the dependent variable and x the independent variable, this relationship is described as the regression of y on x. The relationship can be represented by a simple equation called the regression equation. The regression equation representing how much y changes with any given change of x can be used to construct a regression line on a scatter diagram, and in the simplest case this is assumed to be a straight line.

Procedure :

1. Compute $\sum X, \sum Y, \sum X^2, \sum Y^2$ and $\sum XY$.

2. Calculate correlation coefficient by

$$\rho = \frac{N \sum XY - \sum X \sum Y}{\sqrt{N \sum X^2 - (\sum X)^2} \sqrt{N \sum Y^2 - (\sum Y)^2}}$$

3. Compute $\bar{X} = \frac{\sum X}{N}$ and $\bar{Y} = \frac{\sum Y}{N}$

4. Calculate regression coefficient by

$$b_{yx} = \frac{N \sum XY - \sum X \sum Y}{N \sum X^2 - (\sum X)^2}$$

5. The regression line Y on X is given by

$$Y = b_{yx}(X - \bar{X}) + \bar{Y}$$

6. Plot the given data and the Regression line in a graph.

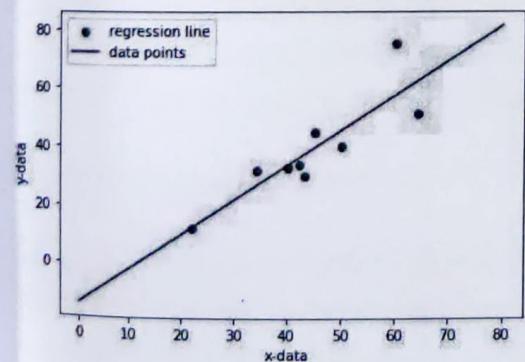
Program :

```
import numpy as np
import math
import matplotlib.pyplot as plt
x=[int(i) for i in input().split()]
y=[int(i) for i in input().split()]
N=len(x)
sx=sy=sxy=sx2=sy2=0
for i in range(0,N):
    sx=sx+x[i]
    sy=sy+y[i]
    sxy=sxy+x[i]*y[i]
    sx2=sx2+x[i]**2
    sy2=sy2+y[i]**2
r=(N*sxy-sx*sy)/(math.sqrt(N*sx2-sx**2)*math.sqrt(N*sy2-sy**2))
print("The coorelation coefficient is %0.3f"%r)
byx=(N*sxy-sx*sy)/(N*sx2-sx**2)
xmean=sx/N
ymean=sy/N
print("The regression line Y on X :::: y = %0.3f %0.3f (x-%0.3f)"%(ymean,byx,xmean))
plt.scatter(x,y)
def Reg(x):
    return ymean+byx*(x-xmean)
x=np.linspace(0,80,51)
y1=Reg(x)
plt.plot(x,y1,'r')
plt.xlabel('x-data')
plt.ylabel('y-data')
plt.legend(['regression line','data points'])
```

Output :

```
60 34 40 50 45 40 22 43 42 64
75 32 33 40 45 33 12 30 34 51
The coorelation coefficient is 0.562
The regression line Y on X :::: y = 38.500 1.186 (x-44.000)
```

```
<matplotlib.legend.Legend at 0x2f2c4267130>
```



Results :

Thus to analyse given data using coefficient of correlation and regression line is successfully completed.

DATE:11/10/22

EXP NO.4

Single server with infinite capacity (M/M/1):(oo/FIFO)

Aim :

To find (a) average number of materials in the system (b) average number of materials in the conveyor (c) waiting time of each material in the system (d) waiting time of each material in the conveyor, if the arrival of materials follow poisson process with the mean interval time 12 seconds, service time of lathe machine follows exponential distribution with mean service time 1 second and average service time of robot is 7seconds.

Software required :

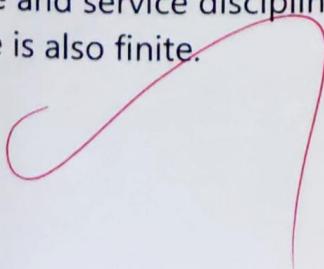
Visual components and Python

Theory:

Queuing are the most frequently encountered problems in everyday life. For example, queue at a cafeteria, library, bank, etc. Common to all of these cases are the arrivals of objects requiring service and the attendant delays when the service mechanism is busy. Waiting lines cannot be eliminated completely, but suitable techniques can be used to reduce the waiting time of an object in the system. A long waiting line may result in loss of customers to an organization. Waiting time can be reduced by providing additional service facilities, but it may result in an increase in the idle time of the service mechanism.

The (M/M/1): (∞ /FIFO) system

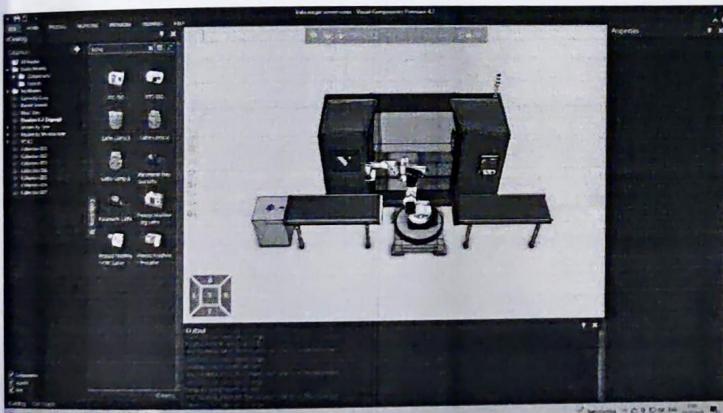
This is a queuing model in which the arrival is Marcovian and departure distribution is also Marcovian, number of server is one and size of the queue is also Marcovian, no. of server is one and size of the queue is infinite and service discipline is 1st come 1st serve(FCFS) and the calling source is also finite.



Procedure :

1. Probability of zero units in the queue (P_0) = $1 - \frac{\lambda}{\mu}$
2. Average queue length (L_q) = $\frac{\lambda^2}{\lambda(\lambda - \mu)}$
3. Average number of units in the system (L_s) = $\frac{\mu}{\lambda - \mu}$
4. Average waiting time of an arrival (W_q) = $\frac{\lambda}{\mu(\lambda - \mu)}$
5. Average waiting time of an arrival in the system (W_s) = $\frac{1}{\lambda - \mu}$

Experiment:



Program

```
arr_time=float(input("Enter the mean inter arrival time of objects  
from Feeder (in secs): "))  
ser_time=float(input("Enter the mean inter service time of Lathe  
Machine (in secs) : "))  
Robot_time=float(input("Enter the Additional time taken for the  
Robot (in secs) : "))  
lam=1/arr_time  
mu=1/(ser_time+Robot_time)  
print("-----  
--")  
print("Single Server with Infinite Capacity - (M/M/1):(oo/FIFO)")  
print("-----  
--")
```

```

print("The mean arrival rate per second : %0.2f "%lam)
print("The mean service rate per second : %0.2f "%mu)
if (lam < mu):
    Ls=lam/(mu-lam)
    Lq=Ls-lam/mu
    Ws=Ls/lam
    Wq=Lq/lam
    print("Average number of objects in the system : %0.2f "%Ls)
    print("Average number of objects in the conveyor : %0.2f "%Lq)
    print("Average waiting time of an object in the system : %0.2f
secs"%Ws)
    print("Average waiting time of an object in the conveyor :
%0.2f secs"%Wq)
    print("Probability that the system is busy : %0.2f "%(lam/mu) )
    print("Probability that the system is empty : %0.2f "%(1-
lam/mu) )
else:
    print("Warning! Objects Over flow will happen in the conveyor")
print("-----")

```

Output

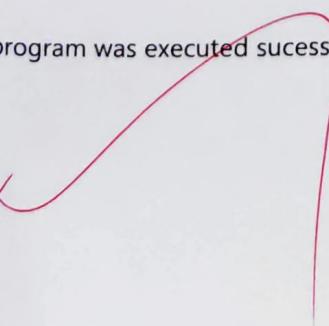
Enter the mean inter arrival time of objects from Feeder (in secs): 12
 Enter the mean inter service time of Lathe Machine (in secs) : 1
 Enter the Additional time taken for the Robot (in secs) : 7

Single Server with Infinite Capacity - (M/M/1):(oo/FIFO)

The mean arrival rate per second : 0.08
 The mean service rate per second : 0.12
 Average number of objects in the system : 2.00
 Average number of objects in the conveyor : 1.33
 Average waiting time of an object in the system : 24.00 secs
 Average waiting time of an object in the conveyor : 16.00 secs
 Probability that the system is busy : 0.67
 Probability that the system is empty : 0.33

Result

Thus the program was executed sucessfully and got the output for single server with infinite capacity..



EXP NO.5
DATE:28/10/22

Multiple server with infinite capacity - (M/M/c):(oo/FIFO)

Aim :

To find (a) average number of materials in the system (b) average number of materials in the conveyor (c) waiting time of each material in the system (d) waiting time of each material in the conveyor, if the arrival of materials follow poisson process with the mean interval time 10 seconds, service time of two lathe machine follow exponential distribution with mean service time 1 second and average service time of robot is 7seconds.

Software required :

Visual components and Python

Theory:

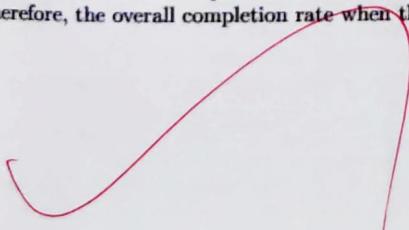
Queueing are the most frequently encountered problems in everyday life. For example, queue at a cafeteria, library, bank, etc. Common to all of these cases are the arrivals of objects requiring service and the attendant delays when the service mechanism is busy. Waiting lines cannot be eliminated completely, but suitable techniques can be used to reduce the waiting time of an object in the system. A long waiting line may result in loss of customers to an organization. Waiting time can be reduced by providing additional service facilities, but it may result in an increase in the idle time of the service mechanism.

M/M/ ∞

Consider a queue that has an infinite number of servers, so that every customer that arrives can immediately enter service, and there is never anyone waiting. We've used the M/M/ ∞ queue as an element in most of our models, but we've usually treated it as a *delay* node rather than an actual queueing model.

We can use a Markov chain to solve for the queue length distribution of the M/M/ ∞ system. Let the Poisson arrival rate be λ and let each of the infinite number of servers have exponential service rate μ .

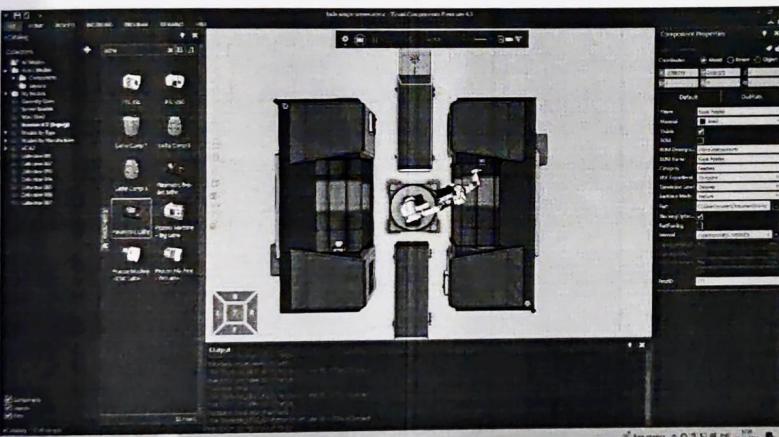
When there are k customers in the system, there are k servers working in parallel. We previously established that the *minimum* of k iid exponential random variables was also exponentially distributed with parameter $k\mu$. Therefore, the overall completion rate when there are k customers in the M/M/ ∞ system is $k\mu$.



Procedure :

1. Traffic intensity $\rho = \frac{\lambda}{c\mu}$
2. Average number of objects in the queue $L_q = \frac{1}{c!} \frac{1}{c} \left(\frac{\lambda}{\mu}\right)^{c+1} \frac{1}{(1-\rho)^2} P_0$,
where $P_0 = \left[\sum_{n=0}^{c-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \frac{1}{c!} \left(\frac{\lambda}{\mu}\right)^c \frac{1}{1-\rho} \right]^{-1}$
3. Average number of objects in the system $L_s = L_q + \frac{\lambda}{\mu}$
4. Average waiting time in the system $W_s = \frac{L_s}{\lambda}$
5. Average waiting time in the queue $W_q = \frac{L_q}{\lambda}$

Experiment:



Program:

```
import math
arr_time=float(input("Enter the mean inter arrival time of objects from Feeder (in secs): "))
ser_time=float(input("Enter the mean inter service time of Lathe Machine (in secs) : "))
Robot_time=float(input("Enter the Additional time taken for the Robot (in secs) : "))
c=int(input("Number of service centre : "))
lam=1/arr_time
mu=1/(ser_time+Robot_time)
print("-----")
print("Multiple Server with Infinite Capacity - (M/M/c):(oo/FIFO)")
print("-----")
print("The mean arrival rate per second : %0.2f "%lam)
print("The mean service rate per second : %0.2f "%mu)
rho=lam/(c*mu)
sum=(lam/mu)**c*(1/(1-rho))/math.factorial(c)
for i in range(0,c):
    sum=sum+(lam/mu)**i/math.factorial(i)
>0=1/sum
```

```

if (rho<1):
    Lq=(P0/math.factorial(c))*(1/c)*(lam/mu)**(c+1)/(1-rho)**2
    Ls=Lq+lam/mu
    Ws=Ls/lam
    Wq=Lq/lam
    print("Average number of objects in the system : %.2f "%Ls)
    print("Average number of objects in the conveyor : %.2f "%Lq)
    print("Average waiting time of an object in the system : %.2f secs"%Ws)
    print("Average waiting time of an object in the conveyor : %.2f secs"%Wq)
    print("Probability that the system is busy : %.2f %(rho))")
    print("Probability that the system is empty : %.2f %(1-rho))")
else:
    print("Warning! Objects Over flow will happen in the conveyor")
print("-----")

```

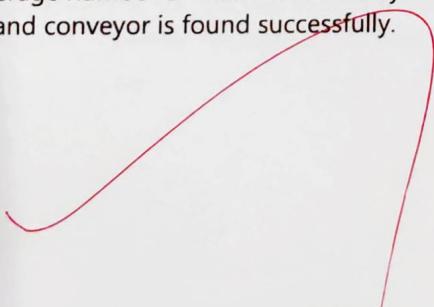
Output:

```

Enter the mean inter arrival time of objects from Feeder (in secs): 10
Enter the mean inter service time of Lathe Machine (in secs) : 1
Enter the Additional time taken for the Robot (in secs) : 7
Number of service centre : 2
-----
Multiple Server with Infinite Capacity - (M/M/c):(oo/FIFO)
-----
The mean arrival rate per second : 0.10
The mean service rate per second : 0.12
Average number of objects in the system : 0.95
Average number of objects in the conveyor : 0.15
Average waiting time of an object in the system : 9.52 secs
Average waiting time of an object in the conveyor : 1.52 secs
Probability that the system is busy : 0.40
Probability that the system is empty : 0.60
-----
```

Result:

Thus the average number of materials in the system and conveyor, waiting time of each material in the system and conveyor is found successfully.



Series Queues with infinite capacity - Open Jackson Network

Aim :

To find (a) average number of materials in the system (b) average number of materials in the each conveyor of (c) waiting time of each material in the system (d) waiting time of each material in each conveyor, if the arrival of materials follow Poisson process with the mean interval time 12 seconds, service time of lathe machine in series follow exponential distribution with service time 1 second, 1.5 seconds and 1.3 seconds respectively and average service time of robot is 7 seconds.

Software required :

Visual components and Python

Theory

Open Jackson Networks

An open Jackson network is a system of k service stations where station i ($i = 1, 2, 3, \dots, k$) has the following characteristics.

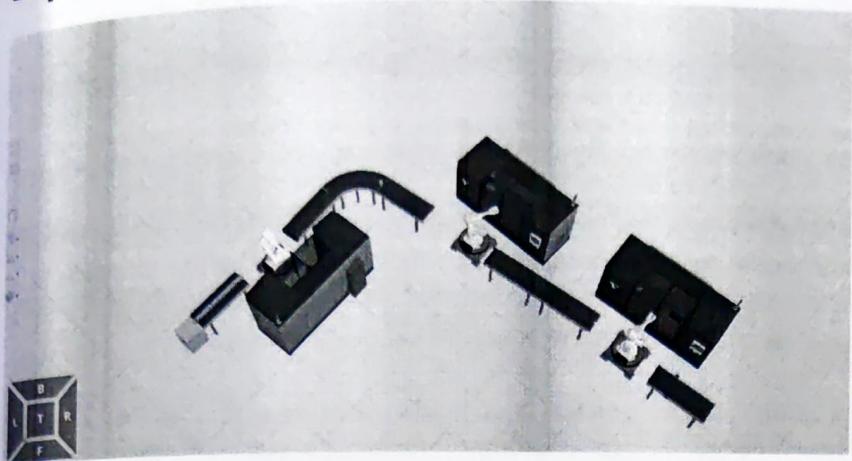
- (i) An infinite queue capacity
- (ii) Customer arrive at station j from outside of the system according to a Poisson processes with parameter r_j .
- (iii) C_i servers at station i with an exponential service time distribution with parameter μ_i
- (iv) Customers completing service at station i next go to station j .
- (v) Let λ_j denote the total arrival rate of customers to the station S_j . Then the traffic flow equation is

$$\lambda_j = r_j + \sum_{i=1}^m \lambda_i p_{ij}$$

Procedure

1. Average number of customers in the system S_j is $L_{S_j} = \frac{\lambda_j}{\lambda_j - \mu_j}$
2. Average number of customers in the overall system $L_S = \sum_{j=1}^k L_{S_j}$
3. Average waiting time in the system $W_S = \frac{L_S}{r_1 + r_2 + \dots + r_k}$

Experiment:



Program

```
arr_time=float(input("Enter the mean inter arrival time of objects from Feeder (in secs): "))
ser_time1=float(input("Enter the mean inter service time of Lathe Machine 1 (in secs) : "))
ser_time2=float(input("Enter the mean inter service time of Lathe Machine 2 (in secs) : "))
ser_time3=float(input("Enter the mean inter service time of Lathe Machine 3 (in secs) : "))
Robot_time=float(input("Enter the Additional time taken for the Robot (in secs) : "))
lam=1/arr_time
mu1=1/(ser_time1+Robot_time)
mu2=1/(ser_time2+Robot_time)
mu3=1/(ser_time3+Robot_time)
print("-----")
print("Series Queues with infinite capacity- Open Jackson Network")
print("-----")
if (lam < mu1) and (lam < mu2) and (lam < mu3):
    Ls1=lam/(mu1-lam)
    Ls2=lam/(mu2-lam)
    Ls3=lam/(mu3-lam)
    Ls=Ls1+Ls2+Ls3
    Lq1=Ls1-lam/mu1
    Lq2=Ls2-lam/mu2
    Lq3=Ls3-lam/mu3
    Wq1=Lq1/lam
    Wq2=Lq2/lam
    Wq3=Lq3/lam
    Ws=Ls/(3*lam)
    print("Average number of objects in the system S1 : %0.2f "%Ls1)
    print("Average number of objects in the system S2 : %0.2f "%Ls2)
    print("Average number of objects in the system S3 : %0.2f "%Ls3)
    print("Average number of objects in the overall system : %0.2f "%Ls)
    print("Average number of objects in the conveyor S1 : %0.2f "%Lq1)
    print("Average number of objects in the conveyor S2 : %0.2f "%Lq2)
    print("Average number of objects in the conveyor S3 : %0.2f "%Lq3)
    print("Average number of objects in the conveyor S1 : %0.2f secs"%Wq1)
    print("Average waiting time of an object in the conveyor S2 : %0.2f secs"%Wq2)
    print("Average waiting time of an object in the conveyor S3 : %0.2f secs"%Wq3)
    print("Average waiting time of an object in the overall system : %0.2f secs"%Wq)
else:
    print("Warning! Objects Over flow will happen in the conveyor")
print("-----")
```

Output

```
Enter the mean inter arrival time of objects from Feeder (in secs): 12
Enter the mean inter service time of Lathe Machine 1 (in secs) : 1
Enter the mean inter service time of Lathe Machine 2 (in secs) : 1.5
Enter the mean inter service time of Lathe Machine 3 (in secs) : 1.3
Enter the Additional time taken for the Robot (in secs) : 7
```

Series Queues with infinite capacity- Open Jackson Network

```
Average number of objects in the system S1 : 2.00
Average number of objects in the system S2 : 2.43
Average number of objects in the system S3 : 2.24
Average number of objects in the overall system : 6.67
Average number of objects in the conveyor S1 : 1.33
Average number of objects in the conveyor S2 : 1.72
Average number of objects in the conveyor S3 : 1.55
Average waiting time of an object in the conveyor S1 : 16.00 secs
Average waiting time of an object in the conveyor S2 : 20.64 secs
Average waiting time of an object in the conveyor S3 : 18.62 secs
Average waiting time of an object in the overall system : 1.52 secs
```

Result :

Thus the experiment executed sucessfully

Completed
P. Jaiswal