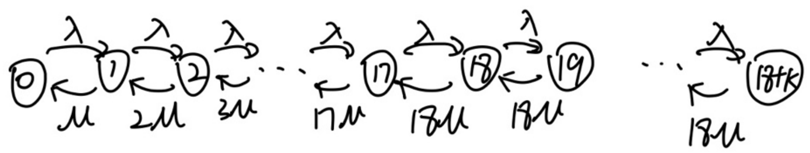
网工数基 M/M/18

学号：　　　　　　　　　　　　　　　　　　　　姓名：

1.



流入 = 流出

0	μP_1 = λP_0		$P_1 = \rho P_0$
1	$\lambda P_0 + 2\mu P_2$ = $\lambda P_1 + \mu P_1$		$P_2 = \frac{\rho^2}{2!} P_0$
2	$\lambda P_1 + 3\mu P_3$ = $\lambda P_2 + 2\mu P_2$		$P_3 = \frac{\rho^3}{3!} P_0$
⋮			
17	$\lambda P_{16} + 18\mu P_{18} = \lambda P_{17} + 17\mu P_{17}$		$P_{18} = \frac{\rho^{18}}{18!} P_0$
18	$\lambda P_{17} + 18\mu P_{19} = \lambda P_{18} + 18\mu P_{18}$		$P_{19} = \frac{\lambda}{18\mu} P_{18}$
⋮			
18+k	$\lambda P_{17+k} + 18\mu P_{19+k} = \lambda P_{18+k} + 18\mu P_{18+k}$		$P_{18+k} = (\frac{\lambda}{18\mu})^k P_{18}$

由 $\sum_{n=0}^{\infty} P_n = 1$，得

$$P_0\left\{1+\rho+\frac{\rho^2}{2!}+\cdots+\frac{\rho^{18}}{18!}+\frac{\rho^{18}}{18!}\left[\frac{\lambda}{18\mu}+\left(\frac{\lambda}{18\mu}\right)^2+\cdots\right]\right\}=1$$

$$P_0=\left(1+\rho+\frac{\rho^2}{2!}+\cdots+\frac{\rho^{18}}{18!}+\frac{\rho^{18}}{18!}\cdot\frac{\lambda}{18\mu-\lambda}\right)^{-1}$$

~~求解时~~，令 $\lambda = 85$，$\mu = 5$　$(5\times 18 = 90)$

2、编程求得 $P_0 = 1.8811 \times 10^{-8}$

3、$L = E[N(t)]$

$$= P_0 \left\{ 1 \cdot \rho + 2 \cdot \frac{\rho^2}{2!} + \cdots + 18 \cdot \frac{\rho^{18}}{18!} + \frac{\rho^{18}}{18!} \left[19 \cdot \frac{\lambda}{18\mu} + 20 \cdot \left(\frac{\lambda}{18\mu}\right)^2 + \cdots \right] \right\}$$

编程求得

$L = 29.6433 \quad T = 0.3487$

对 18个 M/M/1，有

$$L = \frac{\frac{\lambda}{18}}{\mu - \frac{\lambda}{18}} \times 18 = 17 \times 18 = 306$$

$$T = \frac{306}{85} = 3.6$$

对于第二问，求解 P0 时，函数源代码为

```
function [p0] = CalP0(lamda, miu, N)

rou = lamda / miu;
p0 = 1;
tmp = 1;

for i = 1:1:N
    tmp = rou * tmp / i;
    p0 = p0 + tmp;
end

p0 = p0 + tmp * lamda / (N * miu - lamda);

p0 = p0 ^ (-1);

end
```

代码中的参数 lamda 即为λ，miu 即为μ，rou 即为ρ，N 即为服务器个数。

计算 p0 时，首先计算当角标小于 N 时的情况，即第一个 for 循环，此处对 tmp 进行循环，每次乘$\frac{\rho}{i}$，即可得到通项$\frac{\rho^i}{i!}$，累加到 p0 中。当完成第一个 for 循环以后，即可得到 former 部分的值。

之后对 latter 部分，可以得到其级数和为$\frac{\rho^N}{N!}\frac{\lambda}{N\mu-\lambda}$，直接加到 p0 中，最后取-1 次幂，即可得到 p0 的值。

对于第三问，求解 L 与 T 的源代码为

```
function [L, T] = CalLT(lamda, miu, N)

p0 = CalP0(lamda, miu, N);
rou = lamda / miu;
former = 0;
tmp = 1;
```

```
for i = 1:1:N
    tmp = tmp * rou / i;
    former = former + i * tmp;
end

largenumber = 1000000;
latter = 0;
e = 10^(-6);
for i = N+1:1:largenumber
    if(tmp * i * rou / N < e)
        break;
    end
    latter = latter + tmp * i * rou / N;
    tmp = tmp * rou / N;
end

L = p0 * (former + latter);

T = L / lamda;

end
```

参数命名同上，对于前半部分的计算思路和之前也一样，为第一个 for 循环，不过要注意在更新 former 时，要用 i*tmp。

对于 latter 部分，由于无法直接化简级数，故采用“逼近法”来进行求解，设置一个精度以后，当当前项足够小时（小于精度值），即可停止更新。具体在第二个 for 循环中，思路也是根据公式，每次循环都更新 latter 和 tmp(此处 tmp 作为 latter 部分的通项，可以直接在原来 tmp 最后值的基础上进行计算)。

最后根据公式计算 L 和 T 即可。