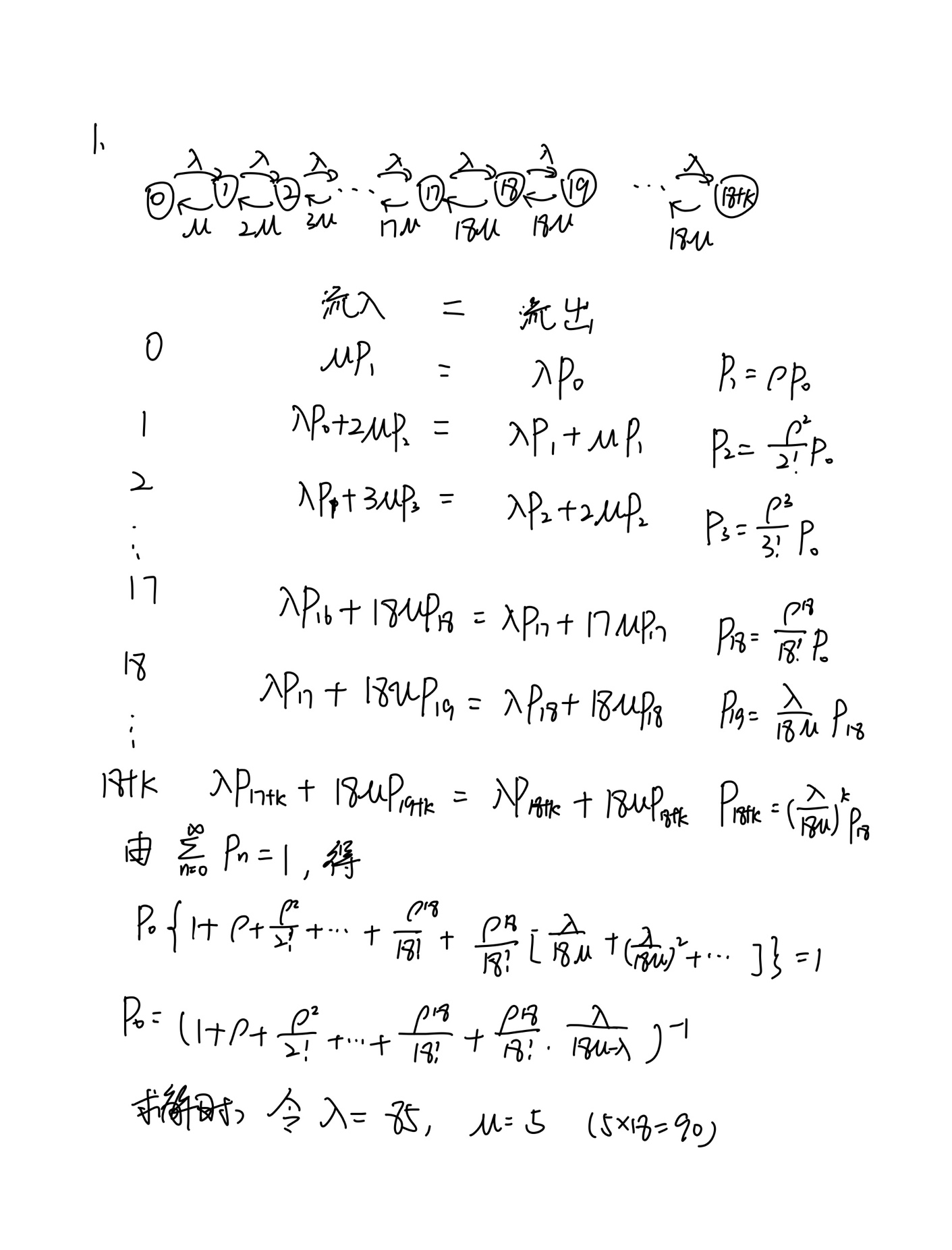
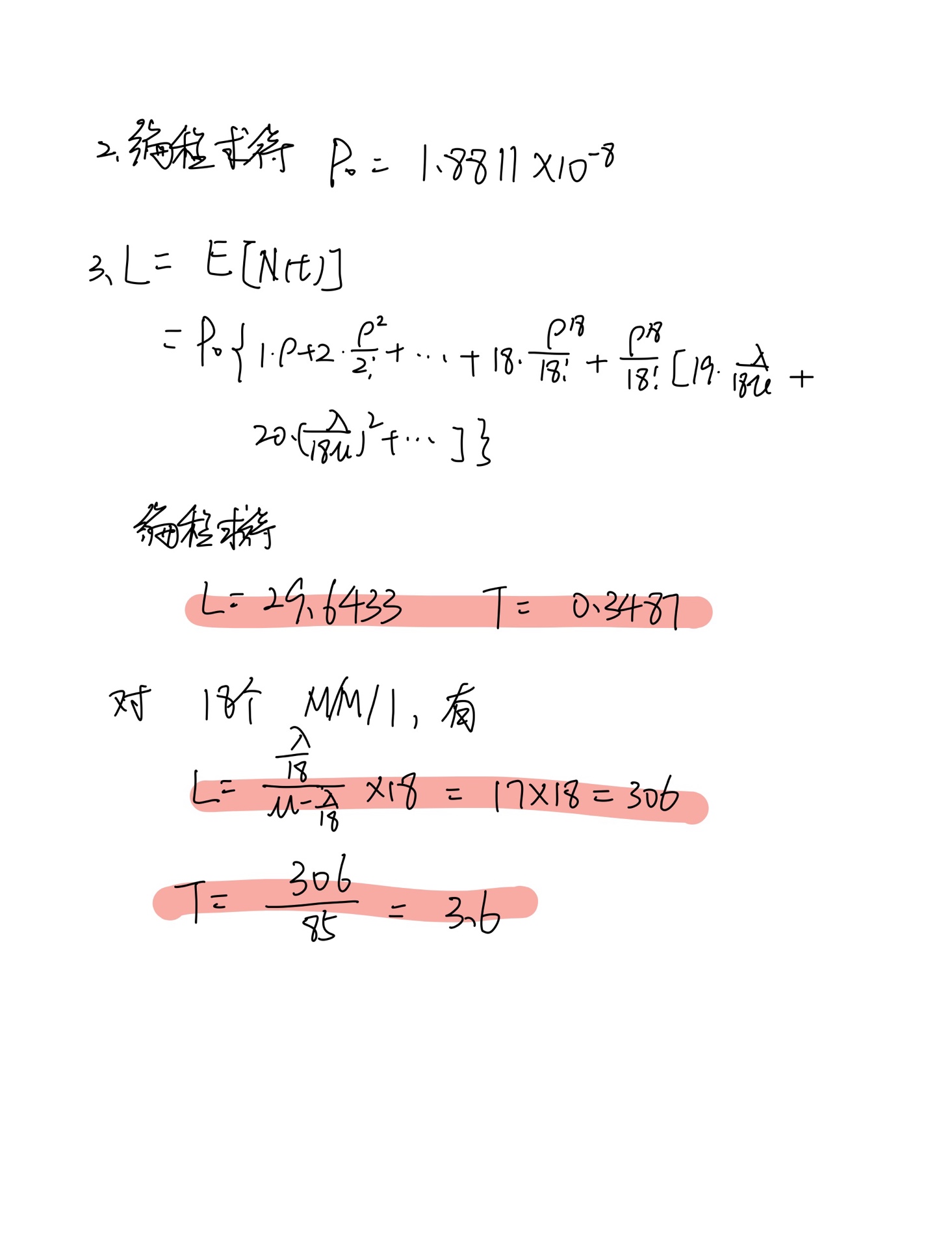
网工数基M/M/18

学号： 姓名：





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，miu即为，rou即为，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，即可得到通项，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，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。