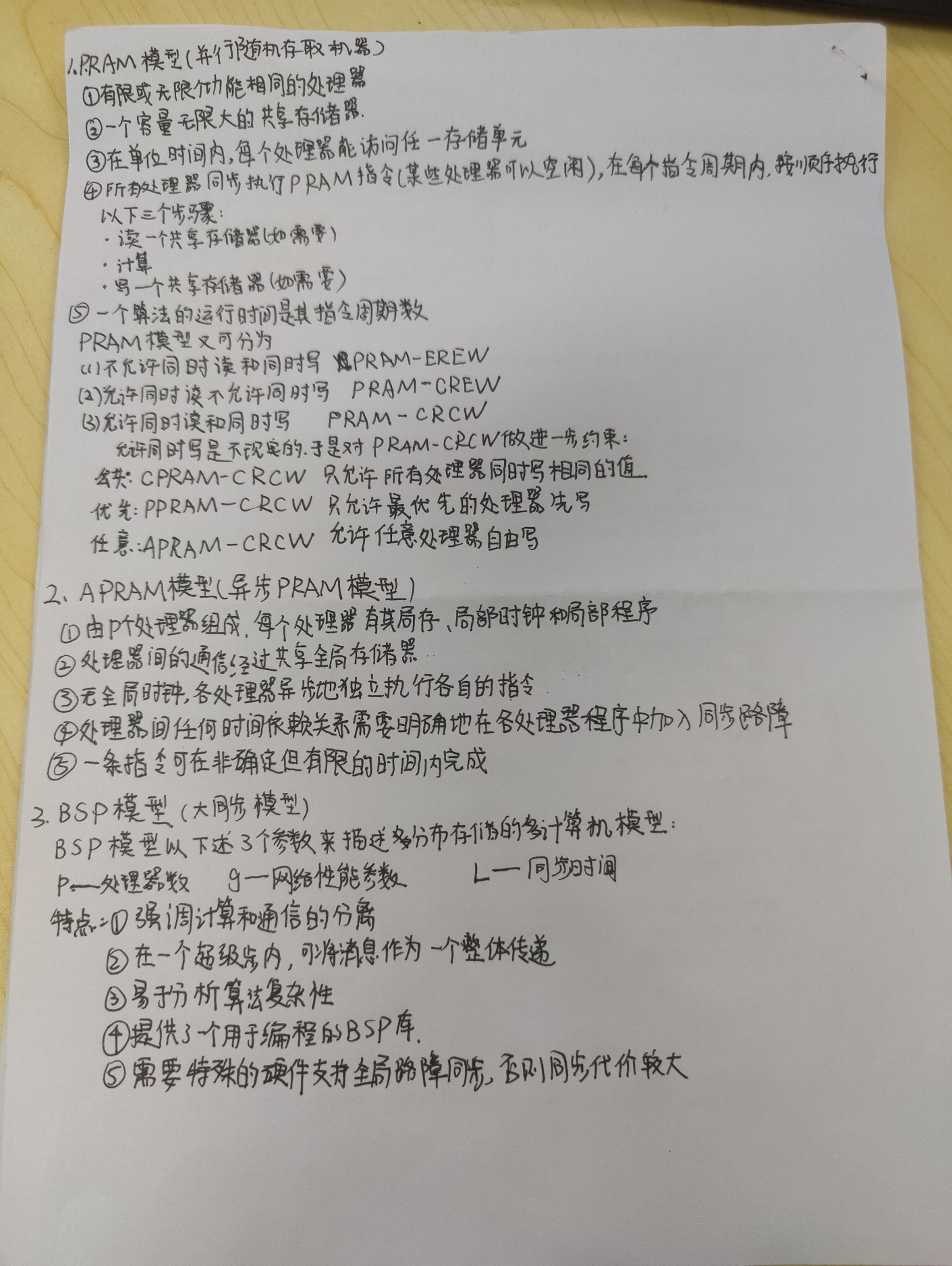
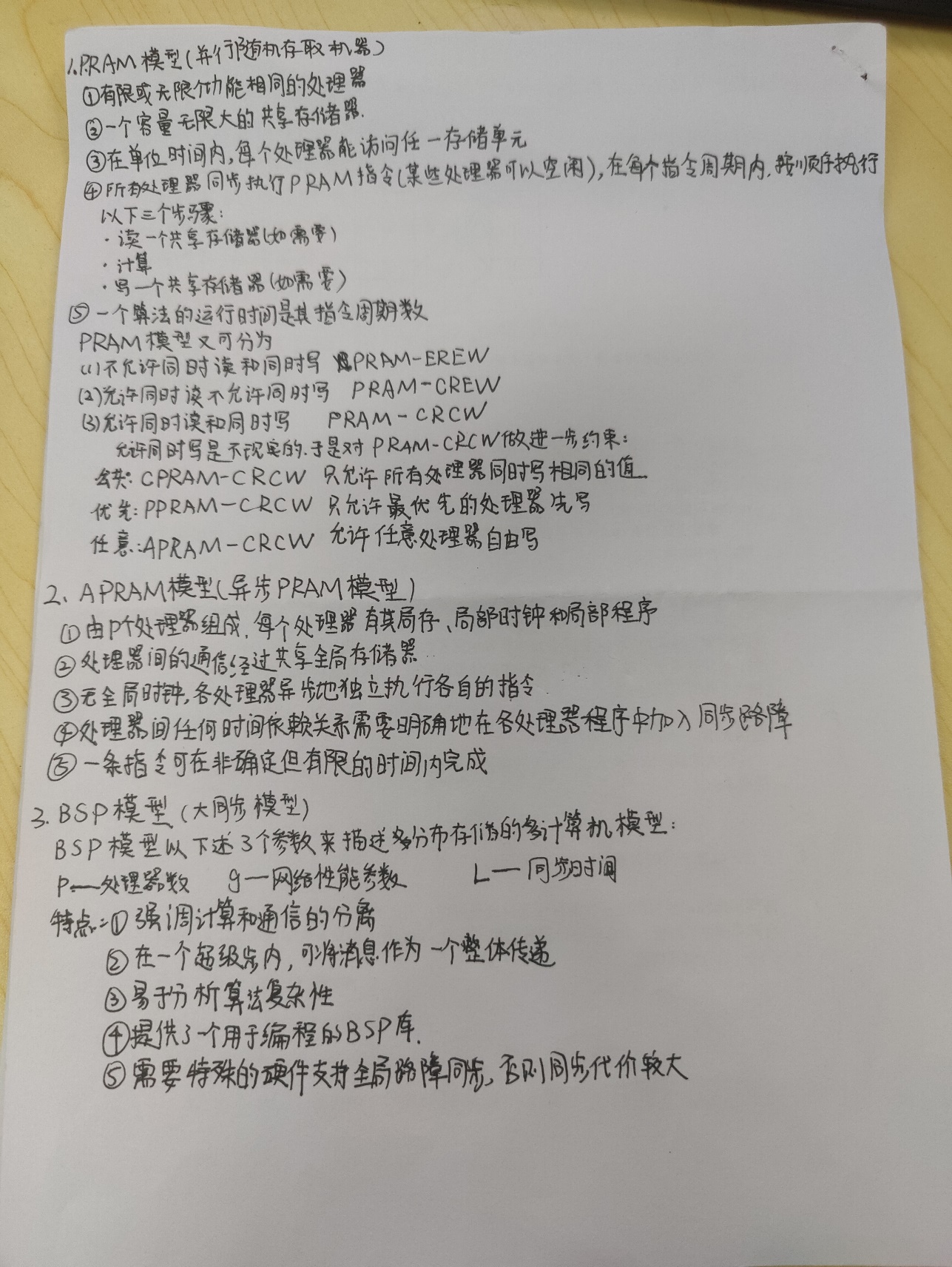
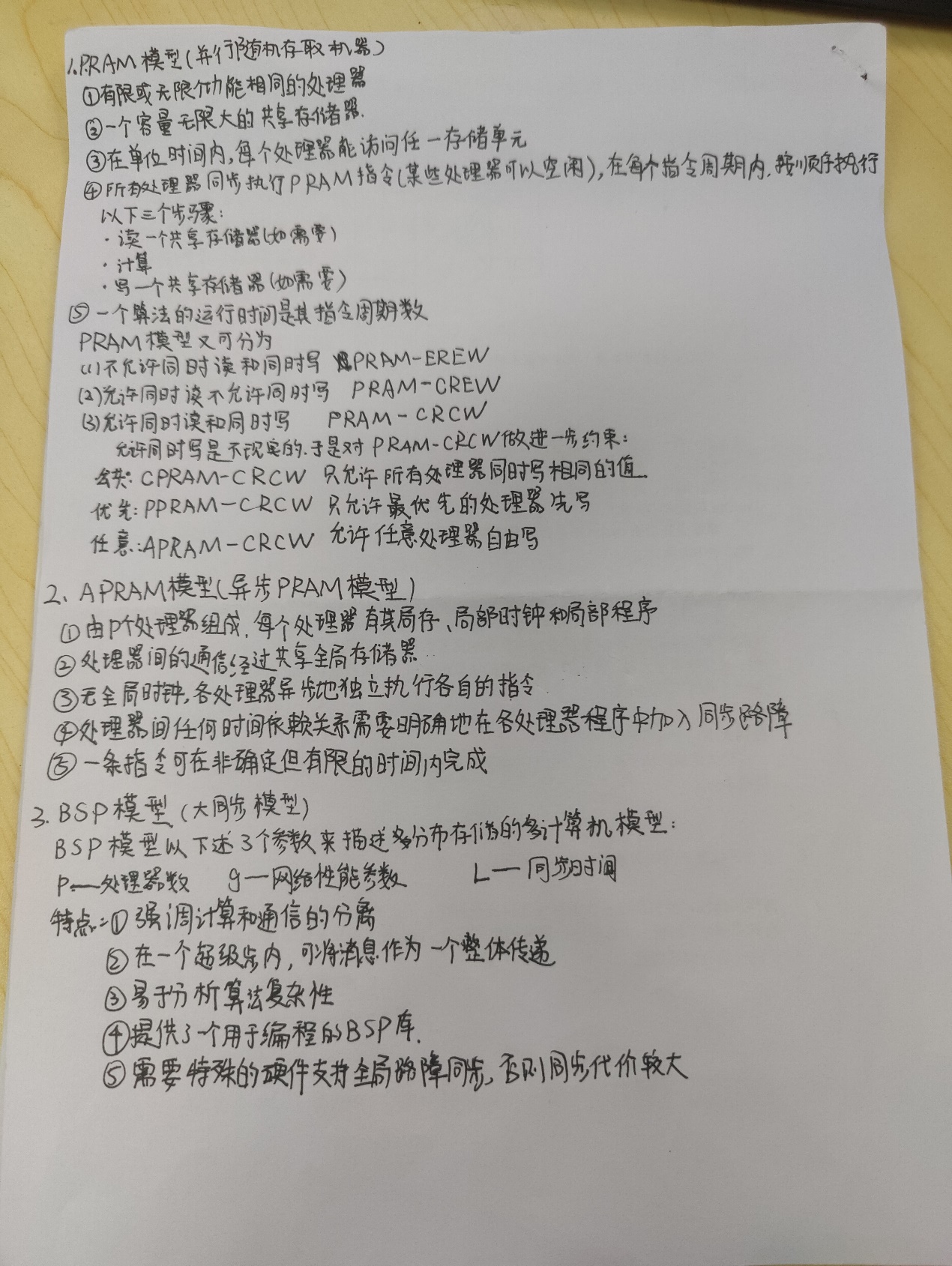
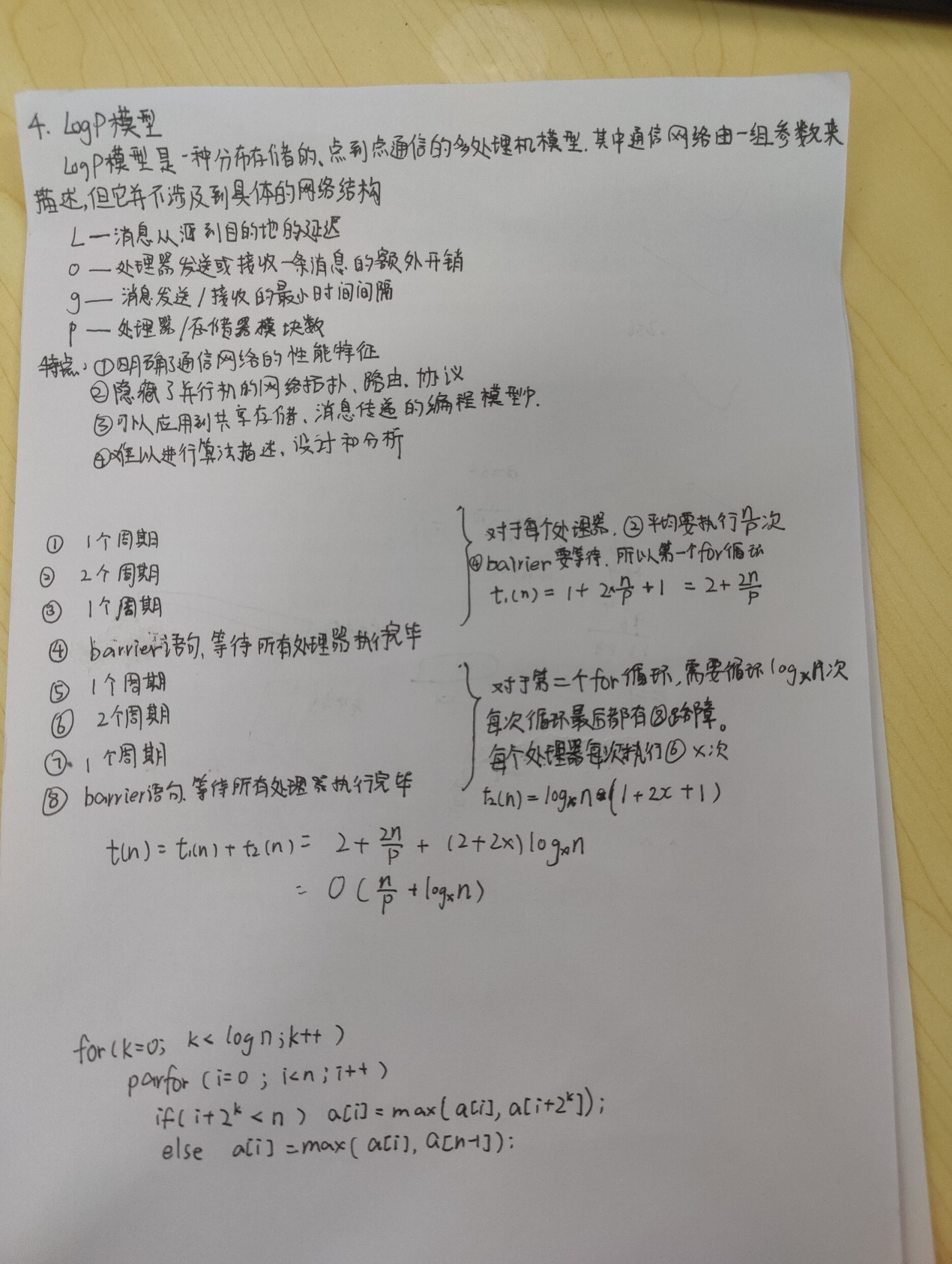
1、简述PRAM模型、APRAM模型、BSP模型和LogP模型的特点。









2、求*n*个数之和的APRAM模型上的算法如下所示，初始时*n*个数存放在数组*a*[0],...,*a*[*n*-1]中，各处理器先求*n*/*p*个数的局部和，然后通过*x*叉树自底向上逐层求和。试分析该算法的运行时间*t*(*n*)，仅需计算有标号的语句。



for all *Pi* where 0≤*i*<*p*

①*ls*=0; //局存

for (*k*=*i*; *k*<*n*; *k*+=*p*)

②*ls*+=*a*[*k*];

③*s*[*i*]=*ls*; //全局存储器

④barrier;

for (*k*=0; *k*<log*xp*; *k*++)

for all *Pi* where 0≤*i*<*p* && *i*%*xk*+1=0

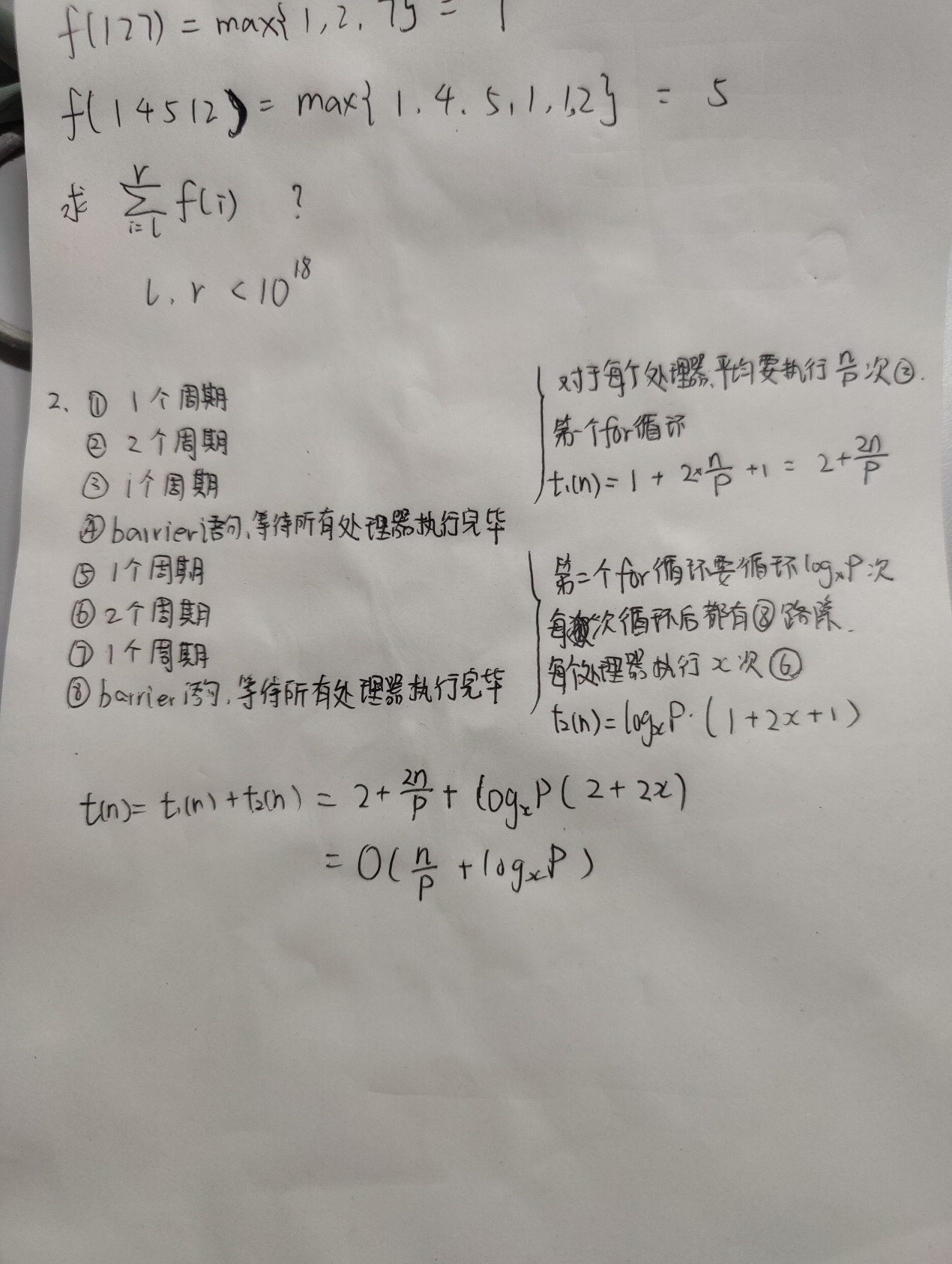
⑤*ls*=0; //局存

for (*j*=*i*; *j*<*i*+*xk*+1 && *j*<*p*; *j*+=*xk*)

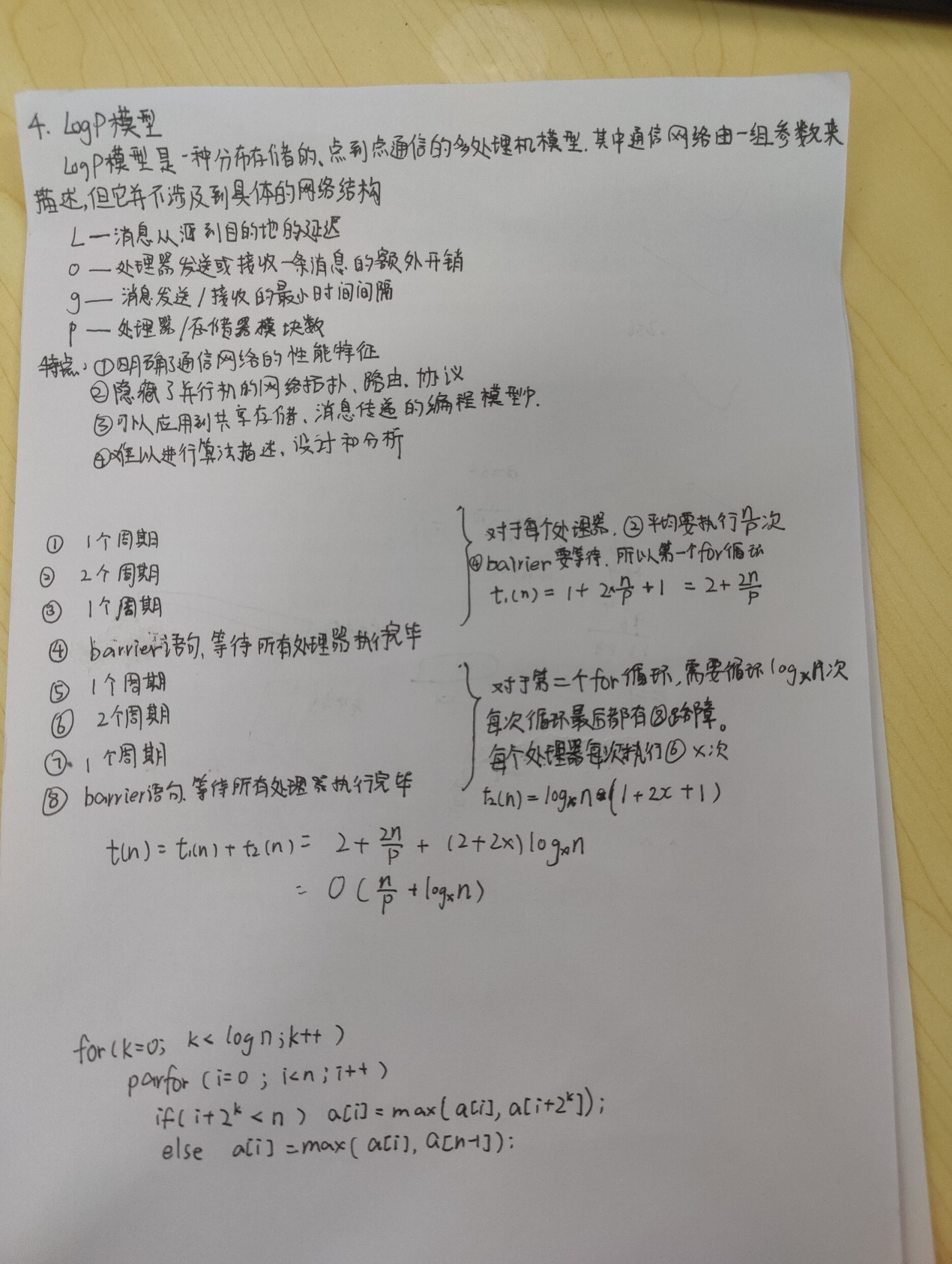
⑥*ls*+=*s*[*j*];

⑦*s*[*i*]=*ls*; //全局存储器

⑧barrier;



3、写出求数组a[n]最大值的PRAM算法，结果存放在a[0]中，第k步后，每个元素存储的是它后面2k个元素之最大值。



4、写出求串匹配次数的PRAM算法，即p[m]在a[n]中出现的次数，要求运行时间为O(logn)，例如，串aba在ababa中出现了2次。

