评分：最终只选第一大题的前两小题，第二大题的前两小题发布。

满分100，4题各25分。

注：第一题只需计算循环体的执行次数(不包括循环变量i,j的增减次数)，不用考虑类型溢出

第一章 概论

1．请计算下面程序中循环语句的执行次数，并给出计算过程（要求是计算准确的次数而不是算法复杂度的阶，其中n为正整数）。

(1)

for (i = 1; i < n - 1; i++)

for (j = n; j >= i; j--)

{

a[i][j] = 1;

}

(2)

i = n \* n;

while (i != 1)

{

i = i / 2;

}

(3)

int i = 0, s = 0;

do {

i = i + 1;

s = s + 10 \* i;

}while ((i<n) && (s<n));

(4)

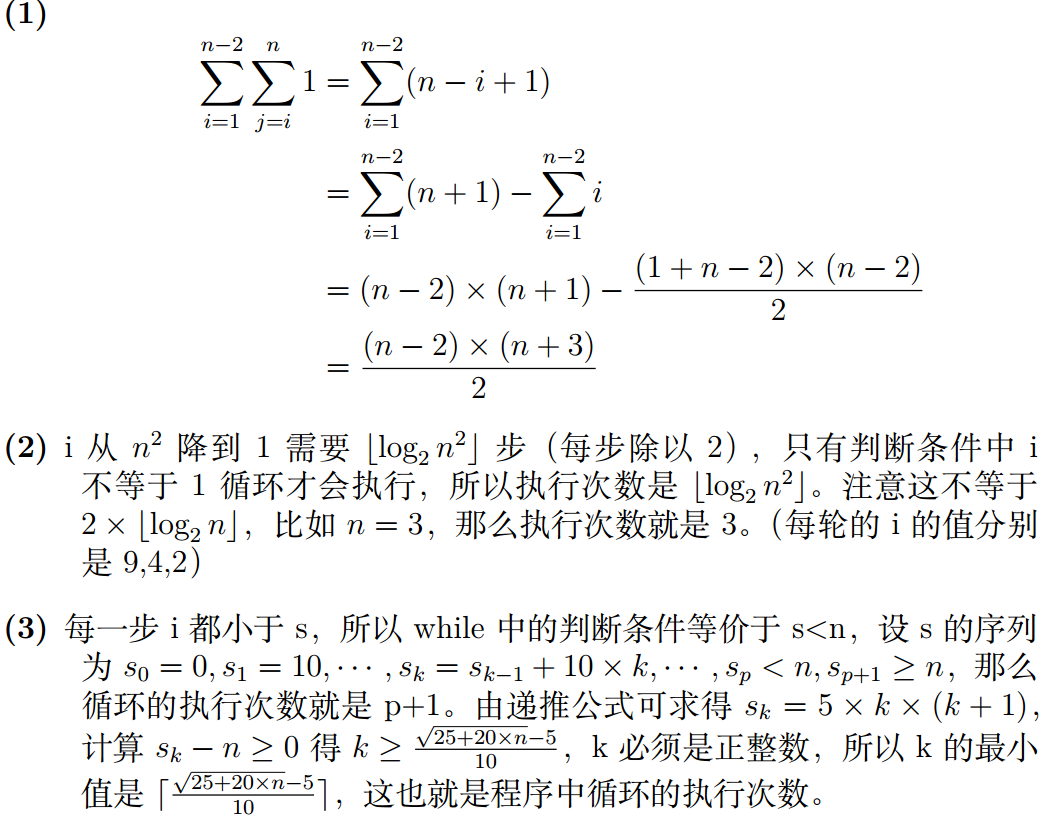
int i = 1;

while(i <= n)

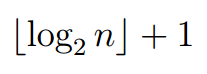
{

i = i \* 2;

}



(4) 答案如下图，注意边界情况



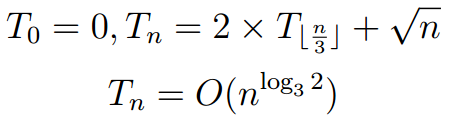
2. 证明

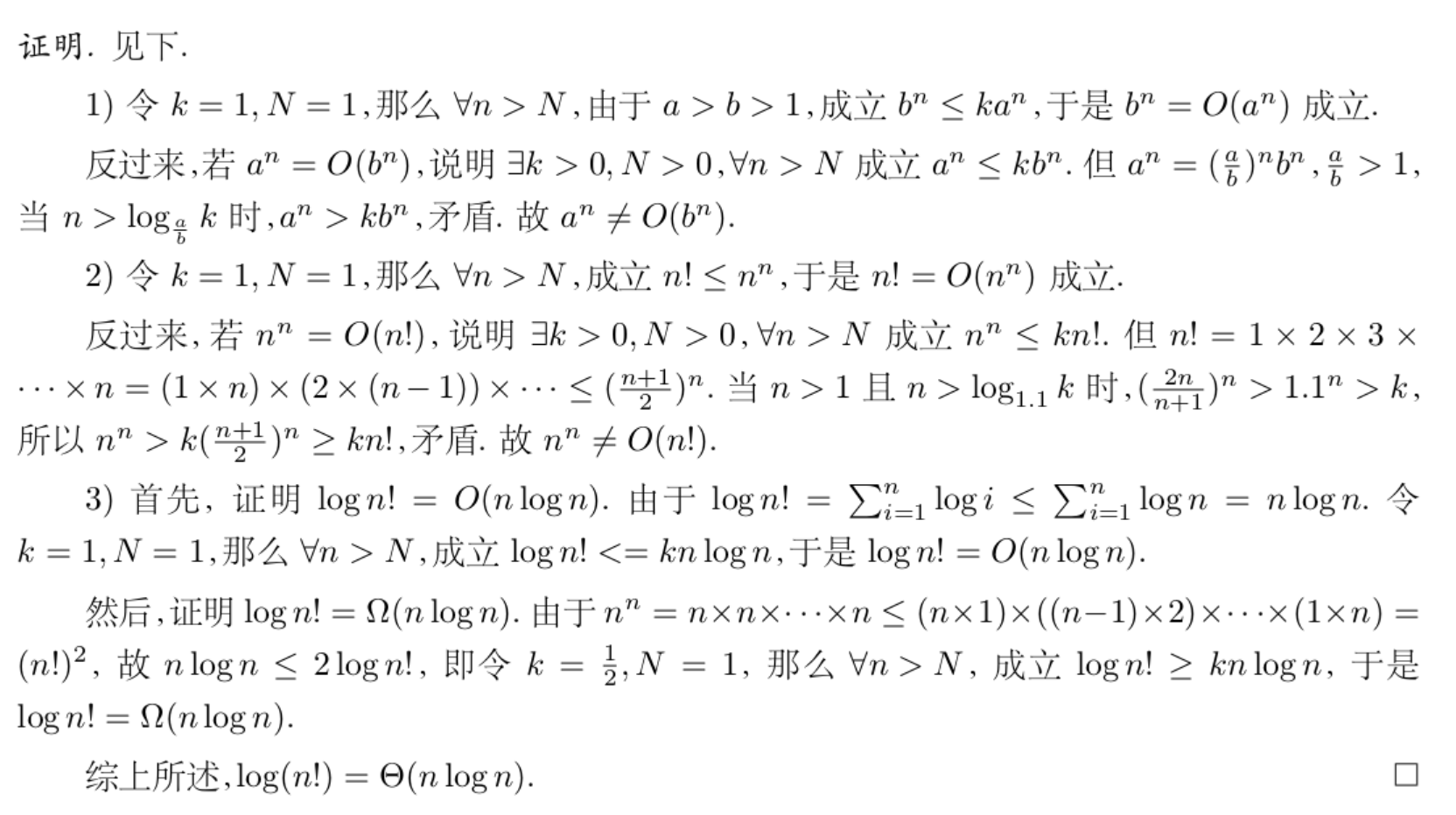
1) 对于任意实数，，但。

2) ，但。

3) 。

4) 如下图，第一行给定Tn的求解公式，证明下图第二行的复杂度上界。





4) 证明如下：

先用数学归纳法证明T(n)单调递增，然后对T(n)进行放缩。因为下取整的数一定小于或等于n/3，而T(n)又是单调递增的，所以T(n)的上界可以由包含T(n/3)的公式限定，T(n/3)又可以放缩得到进一步的上界，直到T(1)或T(0)。对放缩后的式子求和，再进行不等式转换，就可以证明算法的上界。（牵涉到取整时，最大的问题是扩展定义域，否则带着取整符号根本无法计算）（与主定理的联系）

（如果用数学归纳法的思想给出全部证明，那也是可以的：n/3成立推导n一定成立）

