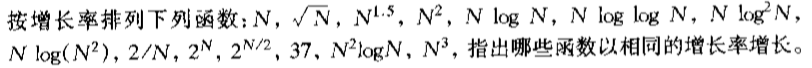
2.1

答案：

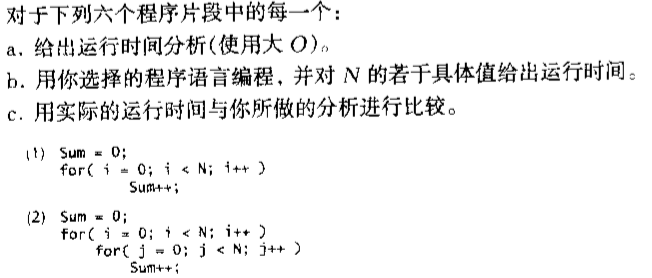


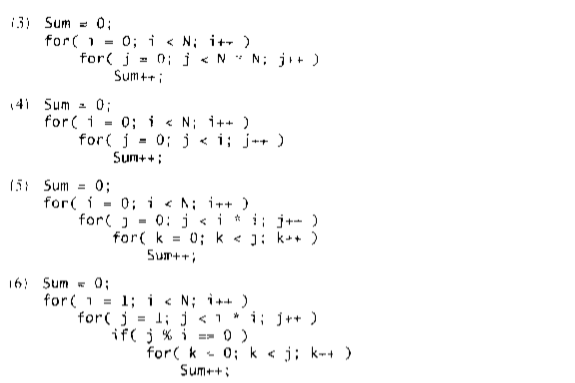
分析：

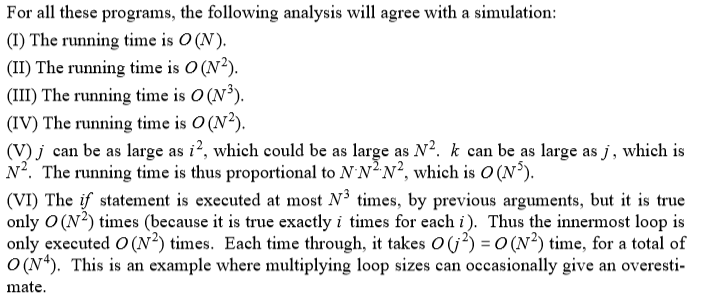
**log(N2)=2log(N)， Nlog(N)与2 Nlog(N)增长率相同 常数不影响增长率**

——————————————

2.6







讲解：

注意

(III)是N3因为内层循环是N2

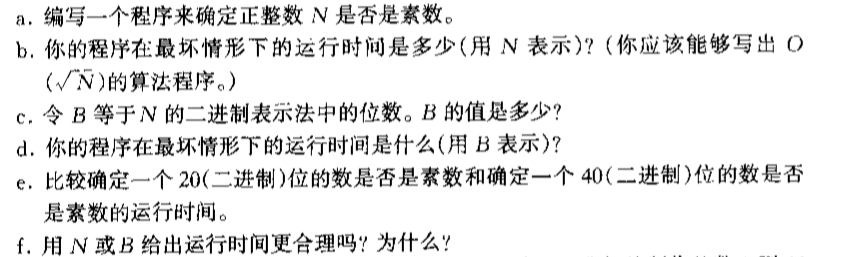
(IV)是N2 因为内层循环的执行次数的边界i是从0到N，平均是N/2 最坏是N，都是N的线性函数

(V)类似上例。第一层循环的复杂度是O(N)；第二层循环的边界i\*i ，平均是(N/2)2，复杂度是O(N2)；第三层循环的边界是j，而j的取值平均是(i\*i/2)即(N/2)2/2 复杂度是O(N2)。三层循环嵌套，是乘法关系，因此总的复杂度O(N5)

(VI)注意：最内层循环的边界j=i2 ，虽然最内循环的执行次数是O(N2) 但每次执行都要跑O(j)次即O(i2)=O(N2) 所以总的是O(N4)。O是上界，考虑的就是最坏情况。

————————————————

2.13





int isPrime(unsigned int N)

{

if (N == 1)

return 0;

if (N % 2 == 0)

return 0;

for (int i = 3; i\*i <= N; i += 2)

{

if (N % i == 0)

return 0;

}

return 1;

}





由二进制转换算法可以推出：B=1+|log2N| 因此O(B)=O(logN); O(N)=O(2B) 因此有答案cd



f) B可以通过位数看出时间复杂度的关系，同一个B可能对应多个N，但就这个算法而言他们的复杂度是一样的，纯粹用N不能看出来。答案是B。