5.3 Suppose $T_1(N) = O(F(N))$ and $T_2(N) = O(F(N))$. Which of the following are true?

a. $T_1(N) + T_2(N) = O(F(N))$
b. $T_1(N) - T_2(N) = O(F(N))$
c. $T_1(N) / T_2(N) = O(1)$
d. $T_1(N) = O(T_2(N))$

5.15 An algorithm takes 0.5 ms for input size 100. How large a problem can be solved in 1 minute (assuming that low-order terms are negligible) if the running time is

a. linear
b. $O(N \log N)$
c. quadratic
d. cubic

5.30 Give an efficient algorithm to determine whether an integer i exists such that $A_i = i$ in an array of increasing integers. What is the running time of your algorithm?

5.21 Occasionally, multiplying the sizes of nested loops can give an overestimate for the Big-Oh running time. This result happens when an innermost loop is infrequently executed. Repeat Exercise 5.20 for the following program fragment:

```
for( int i = 1; i <= n; i++ )
    for( int j = 1; j <= i * i; j++ )
        if( j % i == 0 )
            for( int k = 0; k < j; k++ )
                sum++;
```