Exercises from Section 3.2: (2), (8), (12), (14), (22), (26), (34)

2. Determine whether each of these functions is $O(x^2)$.

a) $f(x) = 17x + 11$
b) $f(x) = x^2 + 1000$
c) $f(x) = x \log x$
d) $f(x) = x^4/2$
e) $f(x) = 2^x$
f) $f(x) = \lfloor x \rfloor \cdot \lceil x \rceil$

8. Find the least integer n such that $f(x)$ is $O(x^n)$ for each of these functions.

a) $f(x) = 2x^2 + x^3 \log x$
b) $f(x) = 3x^5 + (\log x)^4$
c) $f(x) = (x^4 + x^2 + 1)/(x^4 + 1)$
d) $f(x) = (x^3 + 5 \log x)/(x^4 + 1)$

12. Show that $x \log x$ is $O(x^2)$ but that x^2 is not $O(x \log x)$.

14. Determine whether x^3 is $O(g(x))$ for each of these functions $g(x)$.

a) $g(x) = x^2$
b) $g(x) = x^3$
c) $g(x) = x^2 + x^3$
d) $g(x) = x^2 + x^4$
e) $g(x) = 3^x$
f) $g(x) = x^3/2$

22. Arrange the functions $(1.5)^n$, n^{100}, $(\log n)^3$, $\sqrt{n} \log n$, 10^n, $(n!)^2$, and $n^{99} + n^{98}$ in a list so that each function is big-O of the next function.

26. Give a big-O estimate for each of these functions. For the function g in your estimate $f(x)$ is $O(g(x))$, use a simple function g of smallest order.

a) $(n^3 + n^2 \log n)(\log n + 1) + (17 \log n + 19)(n^3 + 2)$
b) $(2^n + n^2)(n^3 + 3^n)$
c) $(n^n + n2^n + 5^n)(n! + 5^n)$

34. a) Show that $3x^2 + x + 1$ is $\Theta(3x^2)$ by directly finding the constants k, C_1, and C_2 in Exercise 33.

b) Express the relationship in part (a) using a picture showing the functions $3x^2 + x + 1$, $C_1 \cdot 3x^2$, and $C_2 \cdot 3x^2$, and the constant k on the x-axis, where C_1, C_2, and k are the constants you found in part (a) to show that $3x^2 + x + 1$ is $\Theta(3x^2)$.

Exercises from Section 3.3: (2), (4), (16), (18), (20), (42)

2. Give a big-O estimate for the number additions used in this segment of an algorithm.

```
t := 0
for i := 1 to n
  for j := 1 to n
    t := t + i + j
```

4. Give a big-O estimate for the number of operations, where an operation is an addition or a multiplication, used in this segment of an algorithm (ignoring comparisons used to test the conditions in the **while** loop).

```
i := 1
t := 0
while i ≤ n
    t := t + i
    i := 2i
```

16. What is the largest n for which one can solve within a day using an algorithm that requires $f(n)$ bit operations, where each bit operation is carried out in 10^{-11} seconds, with these functions $f(n)$?

a) $\log n$
b) $1000n$
c) n^2
d) $1000n^2$
e) n^3
f) 2^n
g) 2^{2n}
h) 2^{2^n}

18. How much time does an algorithm take to solve a problem of size n if this algorithm uses $2n^2 + 2^n$ operations, each requiring 10^{-9} seconds, with these values of n?

a) 10
b) 20
c) 50
d) 100

20. What is the effect in the time required to solve a problem when you double the size of the input from n to $2n$, assuming that the number of milliseconds the algorithm uses to solve the problem with input size n is each of these functions? [Express your answer in the simplest form possible, either as a ratio or a difference. Your answer may be a function of n or a constant.]

a) $\log \log n$
b) $\log n$
c) $100n$
d) $n \log n$
e) n^2
f) n^3
g) 2^n

42. Find the complexity of the greedy algorithm for scheduling the most talks by adding at each step the talk with the earliest end time compatible with those already scheduled (Algorithm 7 in Section 3.1). Assume that the talks are not already sorted by earliest end time and assume that the worst-case time complexity of sorting is $O(n \log n)$.