Section 4.3 – Some answers

17. The algorithm goes from 1 to 2n in steps of 2. So it takes n steps. I.e. \( f(n) = n \).
This means \( f(n) = \Theta(n) \) (this follows from theorem 4.3.4)

18. The inside loop \((j = 1 \text{ to } n)\) takes n steps. The outside loop is means we execute this n times. This means \( f(n) = n \cdot n = n^2 \). This means \( f(n) = \Theta(n^2) \) (this follows from theorem 4.3.4)

21. Similar to problem 18, this algorithm has 3 nested loops each executed n times. This means \( f(n) = n \cdot n \cdot n = n^3 \). This means \( f(n) = \Theta(n^3) \) (this follows from theorem 4.3.4)

23. In this algorithm the number of times steps are executed depend on n,i and j.
Consider some examples:
\( N=1 \) then \( x=x+1 \) is executed once.
\( N=2 \) then \( x=x+1 \) is executed \( 1+ (1+2) \) times.
\( N=3 \) then \( x=x+1 \) is executed \( 1+ (1+2) + (1+2+3) \) times.
\( N=4 \) then \( x=x+1 \) is executed \( 1+ (1+2) + (1+2+3) + (1+2+3+4) \) times.

In general we get the formula \( 1+ (1+2) + (1+2+3) + (1+2+3+4) + \ldots + (1+2+3+4+\ldots+n) \)
Note that:
\[
1 + (1 + 2) + (1 + 2 + 3) + \ldots + (1 + 2 + \ldots + n) = \frac{1 \cdot 2}{2} + \frac{2 \cdot 3}{2} + \frac{3 \cdot 4}{2} + \ldots + \frac{n \cdot (n + 1)}{2}
\]
\[
= \frac{1}{2} \left( 1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \ldots + n \cdot (n + 1) \right) = \frac{1}{2} \left( \frac{n(n+1)(n+2)}{6} \right) = \frac{1}{12} (n^3 + 3n^2 + 2n)
\]
(Using problem 2 in section 1.7). Because \( f(n) \) is polynomial we conclude that \( f(n) = \Theta(n^3) \)