Chapter 2

1. (3) Show using definition of \( \Theta \) that \( \frac{1}{2}n^2 - 5n = \Theta(n^2) \)

\[
\frac{1}{2}n^2 - 5n \leq cn^2 \quad (1)
\]

\[
\frac{1}{2}n^2 - 5n \leq cn^2, \text{ for } c = 1 \text{ and } n_0 > 2 \quad (2)
\]

\[
\frac{1}{2}n^2 - 5n \geq cn^2 \quad (4)
\]

\[
\frac{1}{2}n^2 - 5n \geq cn^2 \text{ for } c = 0.1 \text{ and } n_0 > 11 \quad (5)
\]

2. For the following pair of functions indicate whether \( f(n) \) is \( O, \Omega, \Theta \) of \( g(n) \):

- \( n^k, c^n \) Answer \( n^k = O(c^n) \)
- \( 2^n, 2^{n/2} \) Answer \( 2^n = \Omega(2^{n/2}) \)
- \( n^2, n \log^2 n \) Answer \( n^2 = \Omega(n \log^2 n) \)

3. Chapter 2, Problem 1 c, d

- 100n^2 double the input \( 100(2n)^2 = 400n^2 \). The algorithms will run 4 times slower.
- \( n \log n \) double the input \( 2n \log(2n) = 2n(\log n + \log 2) = 2n \log n + 2 \log 2 \). The algorithms will run 2 times slower.

4. (5) Chapter 2, Problem 2 c, e

- 100n^2 = 60 \times 60 \times 10^{10} \text{ and solve for } n \text{ gives us } n = 6 \times 10^5 .
- \( 2^n = 60 \times 60 \times 10^{10} \text{ and solving for } n \text{ gives us } n = \log_2 3600 \times 10^{10} \text{ this is rounds up to } n = 45 .

5. (5) Consider sorting \( n \) numbers stored in an array \( A \) by first selecting the smallest element and exchanging it with \( A[1] \). The finding a second smallest element and exchanging it with \( A[2] \), an continue for the first \( (n-1) \) elements in the array. Write pseudocode for this algorithm and give the best case and worst-case running time.

Worst case running time for finding a minimum in \( n \) dimensional array is going to take \( n \) steps. Then we have to repeat that operation for the remaining \( n - 1 \) dim array, which is going to take \( n - 1 \) steps, etc. Hence the worst case running time is \( n + (n - 1) + (n - 2) + \ldots + 2 = \Theta(n^2) \)

6. Chapter 2, Problem 3

We know from the chapter that polynomials grow slower than exponentials so we can consider \( f_1, f_2, f_3, f_6 \) as one group and \( f_4 \) and \( f_5 \) as other group. For polynomials we can order them by their exponent so \( f_2 \) is before \( f_3 \) before \( f_1 \). \( f_6 \) grows faster then \( n^2 \) but slower then \( n^c \) for some \( c > 2 \). We can insert \( f_6 \) in this order between \( f_3 \) and \( f_1 \). Exponentials can be ordered by their bases, so we can put \( f_4 \) before \( f_5 \). So the functions in the ascending order of growth rate

\[ \sqrt{2n}, n + 10, n^2 \log n, n^{2.5}, 10^n, 100^n \]