1. Rank the following functions by order of growth from the slowest to the fastest (notation: \(\log n = \log_2 n\))

\[
1000, 10\log n, 4n^2, \quad n^2, \quad 2^n, \quad 100n, \quad 2^{2n}
\]

2. Compare the two functions \(n^2\) and \(2^n/4\) for various values of \(n\). Determine when the second becomes larger than the first.

3. Determine the time complexity of the following code segments. In each case, justify your answer.

   (a) \(\text{sum} = 0;\)
   
   \[
   \text{for}(i=1; i<=2*n; i++) \quad \text{sum} = \text{sum} + 1;
   \]

   (b) \(\text{sum} = 0;\)
   
   \[
   \text{for}(i=1; i<=n^2*n; i++) \quad \text{sum} = \text{sum} + 1;
   \]

   (c) \(\text{sum} = 0;\)
   
   \[
   \text{for}(i=1; i<=n*n; i++) \quad \text{sum} = \text{sum} + 1;
   \]

   (d) \(\text{sum} = 0;\)
   
   \[
   \text{for}(i=1; i<=n; i++) \quad \text{for}(j=1; j<=i; j++) \quad \text{sum} = \text{sum} + i;
   \]

   (e) \(\text{sum} = 0;\)
   
   \[
   \text{for}(i=1; i<=100; i++) \quad \text{for}(j=1; j<=n; j++) \quad \text{sum} = \text{sum} + i;
   \]

   (f) \(\text{sum} = 0;\)
   
   \[
   \text{for}(i=1; i<=n; i++) \quad \text{for}(j=1; j<=n; j*=2) \quad \text{sum} = \text{sum} + 1;
   \]

4. Give the complexity of the functions below using big-O notation

   a. \(5n + n^2 - 2\)
   
   b. \(7\)
   
   c. \(4n + 10\log n + 25\)
   
   d. \(3 + 4\log n\)
   
   e. \(n^2 + n^3 + 10\)

5. (a) Explain how to analyze the running time requirements of (i) a for-loop, (b) a while-loop, and (c) an if-then-else statement. (b) A program's main function consists of two function calls in sequence. The first function that is called has a time complexity of \(O(100\log n)\), and the second function has a time complexity of \(O(n)\). What is the overall time complexity of the program? Justify your answer.