1. Let \( f(n) \) and \( g(n) \) be non-negative functions. Using the basic definition of \( \Theta \)-notation, prove that

\[
\max\{f(n), g(n)\} = \Theta(f(n) + g(n))
\]

2. Show that for any real constants \( a \) and \( b \), where \( b > 0 \),

\[
(n + a)^b = \Theta(n^b).
\]

3. Is \( 2^{n+1} = O(2^n) \)? Justify your answer.

4. Is \( 2^{2n} = O(2^n) \)? Justify your answer.

5. Arrange the following functions according to the asymptotic rate of growth from the lowest to the highest. If any two or more functions have the same asymptotic rate of growth, indicate which.

(Two functions \( f, g \) have the same asymptotic rate of growth if \( f = \Theta(g) \); \( f \) grows asymptotically at least as fast as \( g \) if \( g = O(f) \); \( f \) grows asymptotically (strictly) faster than \( g \) if \( g = O(f) \) and \( g \neq O(f) \))

\[
\begin{align*}
n \\
n^2 + \log n \\
\log n \\
\ln \ln n \\
n^2 \\
n^2 \log n \\
\log^2 n \\
\ln n \\
(\log n) \log n \\
n^3 \\
n^2 \log n.
\end{align*}
\]

(ln \( n \) stands for the natural logarithm of \( n \) and \( \log n \) stands for the base-2 logarithm of \( n \).)

6. For each of the following functions, indicate how much the function’s value will change if its argument is increased fourfold:
a. $\log n$

b. $\sqrt{n}$

c. $n$

d. $n^2$

e. $n^3$

f. $2^n$

7. Prove that $\lg(n!) = \Theta(n\lg n)$.

8. Define

$$f(n) = \begin{cases} 3n - 1 & \text{if } n \text{ is divisible by 3} \\ n^2 & \text{otherwise} \end{cases}$$

and

$$g(n) = \begin{cases} n^2 & \text{if } n \text{ is divisible by 3} \\ 2n & \text{otherwise} \end{cases}$$

Is $f(n) = O(g(n))$, is $g(n) = O(f(n))$? Justify your answer.

9. Define

$$f(n) = \begin{cases} 3n^2 + 1 & \text{if } n \leq 1000 \\ n^{3/2} - n + 4 & \text{if } n > 1000 \end{cases}$$

and

$$g(n) = 2n^{3/2}$$

Is $f(n) = O(g(n))$, is $g(n) = O(f(n))$? Why?

10. Is there an integer $k$ such that $\lceil \log n \rceil! = O(n^k)$? Informally, is the function $\lceil \log n \rceil!$ polynomially bounded? Justify your answer.

11. Provide a simple but accurate estimate of these functions in terms of the big-Theta notation.

a. $n^3 + n^2 \log n$

b. $n \log(\sqrt{n} + 1)$

c. $n^{\log n}$

d. $n^{\log n} + (\log n)^n$

e. $n^3 + (n + n^2 + 3)^{1.5} \log n$

11. You are to store objects with unique integer ID’s from $[0..N - 1]$. You expect never to have to store more than $M'$ objects. How would you design the hash function of the type discussed in class? Describe a general approach to design the hash function in terms of $N$ and $M'$. For $N = 100000$ and $M' = 50$, give a concrete example of a hash function $h$ of the type discussed in class. What is $h(63455)$?

12. In the dictionary problem, the goal is to maintain a collection of records, each identified by an ID (you may assume that IDs are non-negative integers from some large range $[0..N - 1]$) and that the number of records in the collection, say $n$, is much smaller than $N$). The records are to be maintained so that one could access a record with a given ID, insert a record with a given ID and delete a record with a given ID.

In class, we discussed three methods to implement a solution to the dictionary problem:
a. Maintain the collection of records as a linked list

b. Maintain the collection of records as entries in the array of size \( N \) (the entry \( i \) in the array stores the record with the ID \( i \), if that record is in the collection, otherwise, the content of the entry is \( null \)).

c. Maintain the collection of records in the hash table (array) of size only slightly larger than the size of the collection (with the entry \( i \) in the hash table storing all records in the collection whose IDs have hash value equal to \( i \)).

For each of the methods point out its advantages and disadvantages (in terms of the performance of access, insert and delete methods, and in terms its storage requirements). Based on that, which approach would you choose if you were to implement the dictionary problem? Why?

13. What is \( 345^{1753} \mod 11 \)?

14. Is \( 7^{1009} - 4^{1009} \) divisible by 17?

15. Select two distinct primes each with 6 binary digits and use them to design an RSA cryptosystem. Produce a public key and the corresponding private key. Encode the message 101110 and then decode it.

16. Describe how to design an RSA cryptosystem based on primes \( p = 13 \) and \( q = 7 \). That is, propose the public key and the private key. Encode the message 100101 and then decode it. Carefully explain all the steps.

17. Describe the extended Euclid algorithm (give pseudocode). Describe what it computes. Show how it works on the input \( x = 72 \) \( y = 33 \).

18. Find the multiplicative inverse of:

   a. 9 modulo 65
   b. 33 modulo 65
   c. 25 modulo 65.

or prove that the multiplicative inverse does not exist.

19. Describe the division algorithm. Describe what it computes. Show how it works on the input \( x = 694 \) \( y = 33 \).

20. Describe the modular exponentiation algorithm. Show how it works to compute \( x^y \mod N \), where \( x = 14 \), \( y = 83 \), \( N = 25 \). All steps must be explained, calculations in each step can be done with the help of a calculator.

21. Show that the number of bits in a binary representation of a positive integer \( n \) is \( \lceil \log(n + 1) \rceil \).

22. Describe an algorithm that for a given \( n \)-bit non-negative integer \( x \) computes \( \lfloor \sqrt{x} \rfloor \). Show that your algorithm is correct. Derive a tight asymptotic bound on the performance of your algorithm.
23. Describe an algorithm that for a given \(n\)-bit non-negative integer \(x\) computes \(\lfloor \log_5 x \rfloor\). Show that your algorithm is correct. Derive a tight asymptotic bound on the performance of your algorithm.

24. Analyze the running time of these lines of code and describe it in the \(O()\) or \(\Theta()\) notation in terms of a function in \(n\), as appropriate, where \(n\) is a non-negative integer specified earlier. Determine those cases (a) - (d), where \(\Theta()\) estimate cannot be used.

(a) \(a = 0\)
    \[\text{for } i = 1 \text{ to } n:\]
    \[a = a + i\]

(b) \(a = 0\)
    \[\text{for } i = 1 \text{ to } n:\]
    \[\text{if } \text{odd:}\]
    \[a = a + i\]

(c) \(a = 0\)
    \[\text{for } i = 1 \text{ to } n:\]
    \[\text{if } i^2 \leq n:\]
    \[\text{for } j = 1 \text{ to } n:\]
    \[a = a + 1\]
    \[\text{else:}\]
    \[a = a - 1\]

(d) \(m = n\)
    \[a = 0\]
    \[\text{while } m > 1:\]
    \[a = a + 1\]
    \[m = m/2\]

24. Use the master theorem to give tight asymptotic bounds for the solutions to the following recurrences:

(a) \(T(n) = 9T(n/3) + n\)
(b) \(T(n) = 9T(n/3) + n^2\)
(c) \(T(n) = 9T(n/3) + n^3\)

25. Find tight asymptotic bounds for the solution to the following recurrence relation:

\[T(n) = 9T(n/3) + n^2 \log n\]
26. Solve the recurrence

\[ T(n) = 2T(\sqrt{n}) + 1 \]

by changing variables.

27. Use the recursion tree method to give an asymptotically tight solution to the recurrence

\[ T(n) = T(n - a) + T(a) + cn, \]

where \( a \geq 1 \) and \( c > 0 \) are constants.