VII. Algorithm Complexity (Chap. 7)

Measuring the Efficiency of Algorithms: (§ 7.4)

1. What to measure?
   Space utilization: amount of memory required
   Time efficiency: amount of time required to process the data.
   — Depends on many factors: size of input, speed of machine, quality of source code, quality of compiler.

Since most of these factors vary from one machine/compiler to another, we count the number of times instructions are executed. Thus, we measure computing time as:

\[ T(n) = \text{the computing time of an algorithm for input of size } n \]
\[ = \text{the number of times the instructions are executed.} \]

2. Example: See ALGORITHM TO CALCULATE MEAN on page 350

/* Algorithm to find the mean of \( n \) real numbers.
   Receive: An integer \( n \geq 1 \) and an array \( x[0], \ldots, x[n-1] \) of real numbers
   Return: The mean of \( x[0], \ldots, x[n-1] \)

----------------------------------------------------------------------------------------*/

1. Initialize \( sum \) to 0.
2. Initialize index variable \( i \) to 0.
3. While \( i < n \) do the following:
   a. Add \( x[i] \) to \( sum \).
   b. Increment \( i \) by 1.
4. Calculate and return \( mean = \frac{sum}{n} \).

\[ T(n) = \] ______________

3. Definition of "big-O notation": The computing time of an algorithm is said to have order of magnitude \( f(n) \), written \[ T(n) = O(f(n)) \]
   if there is some constant \( C \) such that

   ______________________________________________________

   We also say, the complexity of the algorithm is \( O(f(n)) \).

   Example: For the Mean-Calculation Algorithm:

   \[ T(n) = \] ______________

4. The arrangement of the input items may affect the computing time. For example, it may take more time to sort a list of element that are nearly in order than for one that are completely out of order. We might measure it in the best case or in the worst case or try for the average. Usually best-case isn't very informative, average-case is too difficult to calculate; so we usually measure worst-case performance.
5. Example:
   a. LINEAR SEARCH ALGORITHM on p. 354
      
      Worst case: ________________________:
      
      $T_L(n)$ is __________

   b. BINARY SEARCH ALGORITHM on p. 355
      
      Worst case: Item not in the list:
      
      $T_B(n) =$ __________

6. Commonly-used computing times:

   $O(\log_2 \log_2 n)$, $O(\log_2 n)$, $O(n)$, $O(n \log_2 n)$, $O(n^2)$, $O(n^3)$, and $O(2^n)$

   See the table on p. 356 and graphs on p. 357 for a comparison of these.

7. Computing times of Recursive Algorithms

   Have to solve a recurrence relation.

   Example: Towers of Hanoi

   ```c
   void Move(int n,
              char source, char destination, char spare)
   {
      if (n <= 1)                      // anchor
         cout << "Move the top disk from " << source
               << " to " << destination << endl;
      else                                // inductive case
         {                                // inductive case
            Move(n-1, source, spare, destination);
            Move(1, source, destination, spare);
            Move(n-1, spare, destination, source);
         }
   }
   
   T(n) = __________
   ```