1. Let \( a \) and \( b \) be positive integers satisfying \( 1 \leq a < b \), and define \( T(n) \) by

\[
T(n) = \begin{cases} 
    a & 0 \leq n < b \\
    T(n-b)+n+a & n \geq b
\end{cases}
\]

Use the iteration method to find the exact solution to this recurrence.

2. Use the Master Theorem to find tight asymptotic bounds on the following recurrences.

   a. \( T(n) = 3T(2n/3) + n^3 \)
   
   b. \( T(n) = 2T(n/3) + \sqrt{n} \)
   
   c. \( T(n) = 5T(n/4) + n^{\log_2 5} \)
   
   d. \( T(n) = aT(n/b) + n^k \) where \( \log_b(a) < k \).

3. Consider the following algorithm:

   \[A(n)\] (Precondition: \( n \) is a positive integer)
   1. if \( n < 2 \)
   2. return
   3. else
   4. count \( \leftarrow 0 \)
   5. for \( i \leftarrow 1 \) to \( 8 \)
   6. \( A(\lfloor n/2 \rfloor) \)
   7. for \( i \leftarrow 1 \) to \( n^3 \)
   8. count \( \leftarrow \) count + 1

For purposes of analysis, define lines 4 and 8 to be basic operations. Let \( T(n) \) be the number of basic operations performed when the input is \( n \).

   a. Write a recurrence for \( T(n) \).

   b. Find a tight asymptotic bound on \( T(n) \) by any method.
4. Design an algorithm called TernarySearch($A, p, r, t$) which searches for a target $t$ in the sorted subarray $A[p \cdots r]$, returning the appropriate index if $t$ is found, and 0 if $t$ is not found. TernarySearch will work by dividing the input array $A[p \cdots r]$ into three subarrays of approximately equal length, and calling itself recursively on each subarray.

   a. Write pseudo-code for TernarySearch, and prove its correctness. (i.e. formulate a theorem, and prove it by induction on the length $m = r - p + 1$ of $A[p \cdots r]$.)

   b. Write down the recurrence for the run time of TernarySearch and find a tight asymptotic bound on its solution.

5. **The Towers of Hanoi Problem.** Consider three vertical rods, one of which is threaded with $n$ rings of differing sizes. The rings are initially threaded in order of size, with the largest ring at the bottom and the smallest ring at the top. Your task is to transfer all the rings to one of the other two rods. The only operation you may use is to transfer one ring at a time to another rod, with the restriction that no ring is ever placed on top of a smaller one. If $n = 1$ the problem is easy; just move the ring. If $n = 2$, you place the top ring on an intermediate rod, place the second ring on the destination rod, then move the small ring from the temporary rod to the destination rod. If $n = 3$, one can move the rings in the following order:

   1 $\to$ 3
   1 $\to$ 2
   3 $\to$ 2
   1 $\to$ 3
   2 $\to$ 1
   2 $\to$ 3
   1 $\to$ 3

   a. Write a divide and conquer algorithm called Hanoi($n, i, j$) which prints out a set of instructions (as above) to move a stack of $n$ rings from rod $i$ to rod $j$, where $i, j \in \{1,2,3\}$. Prove the correctness of your algorithm.

   b. Write a recurrence for the run time of your algorithm, taking the print instruction as a barometer operation (i.e. $T(n)$ should be the number of print instructions performed for $n$ rings.) Solve this recurrence exactly.