1) For 29-cents using the set of coin types \{1, 5, 10, 12, 25, 50\}, determine the value of fewestCoins[29].

```
   29
  28 24 19 17  
  4  2  4  2  4 
```

2) The recursive definition of the Fibonacci sequence \((0, 1, 1, 2, 3, 5, 8, 13, \ldots)\) is:

\[
\begin{align*}
  f_0 & = 0 \\
  f_1 & = 1 \\
  f_n & = f_{n-1} + f_{n-2} \quad \text{for } n \geq 2
\end{align*}
\]

Complete the recursive algorithm \texttt{fib} that calculates the \(n\)th element in the sequence.

```java
int fib( int n ) {
    if (         ) { return
    } else if (       ) { return
    } else { return
    } // end if
} // end fib
```

3) Complete the recursion tree showing the calls for \texttt{fib}(5).
4) What type of algorithmic problem-solving technique (greedy, divide-and-conquer, dynamic programming) is recursive `fib`?

5) Executing `fib` on large $n$ values takes a long time. What makes the recursive Fibonacci calculation so slow?

6) Can you think of a nonrecursive/iterative algorithm that would be faster?

7) What type of algorithmic problem-solving technique (greedy, divide-and-conquer, dynamic programming) is your iterative algorithm?

8) How much memory space is needed for your algorithm?

9) Can you reduce the amount of memory space needed for your algorithm?