Jonathan Rittmayer

Mat: 30001299

3.2)

A) Instantiate 3 arrays of type Long Int with a length of n

Arr A that holds large integer a's digits as elements

Arr B that holds large integer b's digits as elements

Arr C that has all elements equal to 0

- Multiply element one of Arr A and element one of Arr B

  if the result is greater than 9 add the 10th place digit to the next element in C.

- Repeat the multiplication for all digits of A by the first element of B, and also repeat the carry system for results > 9

- Shift one bit in elements in Arr C and continue the same steps as above for the next digit in B.

- The result in Arr C will be the bit product of the two Long ints A and B.

B) Instantiate 4 arrays of type Long Int with a length of $2 * \left(\frac{n}{2} \text{digit}\right)$

Arr AHeight, ALength

Arr B Height, BLength

Instantiate 4 more arrays of Type Long Int with a length of $4 * \left(\frac{1}{2} \text{digit}\right)$

Arr CHeightTwo, CHeightLength

Arr CLengthTwo, CLengthHeight

- CLengthTwo = ALength * BLength

  if this results in a digit > 9, take $\frac{\text{result}}{10}$ and place it into

  CLengthHeight and place result % 10 in the last location of CLengthTwo.

B continued)

- CLength Height += ALength * BHeight +
                    AHeight * BLength

   For the last bit in CLength Height repeat the $\frac{result}{10}$ & result%i
   from before

- CLength Height += AHeight * BHeight
   and do the same as before and repeat $\frac{result}{10}$ & result%10

C)   The recurrence for the time complexity of the Divide &
     Conquer algorithm from ⓑ is $T(n) = 4T(\frac{n}{2}) + O(n)$

D) Recursion tree Method

$$Total = n(1+2+4+8+...)$$
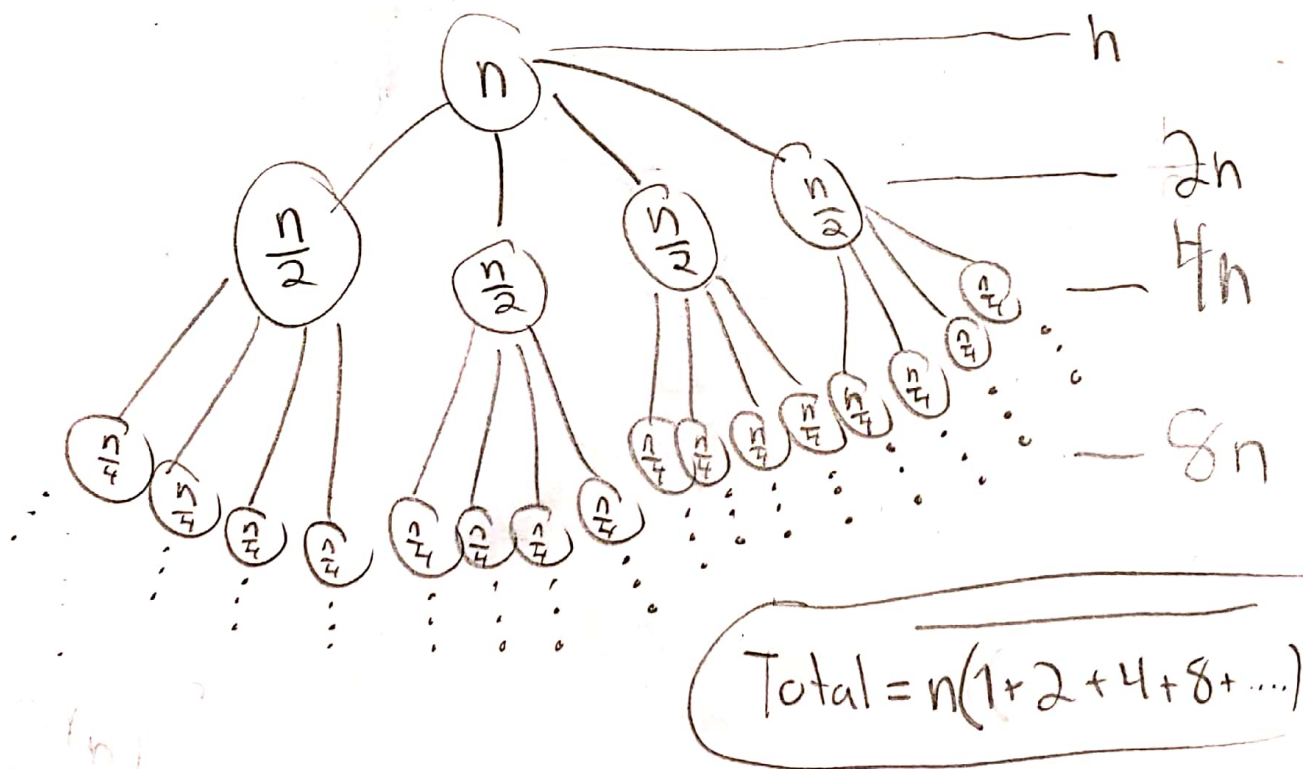
E) Master Theorem $4T(\frac{n}{2}) + O(n)$
   $a = 4$ $n^{\log_b a} = n^{\log_2 4} = n^2$
   $b = 2$
   $f(n) = n$ ← the solution