Homework #2 Key Part A

1.7

Assume we want to multiply the n-bit number x by the m-bit number y. The algorithm must terminate after m recursive calls, because at each call y is halved (the number of digits is decreased by one). Each recursive call requires a division by 2 and multiplication by 2 (both $O(n)$ shifts) and a possible addition of x to the current result (which takes $O(n)$ time). Thus, the total is $O(m \cdot n)$ time.

1.25

Since 127 is prime, by Fermat's Little Theorem, $2^{126} \equiv 1 \pmod{127}$. Then $2 \cdot 2^{125} \equiv 1 \pmod{127}$ and we are looking for a number that, when multiplied by two, is congruent to (or has a remainder of) $1 \pmod{127}$. Since $2 \cdot 64 \equiv 1 \pmod{127}$ and $2 \cdot 2^{125} \equiv 1 \pmod{127}$, then $2^{125} \equiv 64 \pmod{127}$. So the answer is 64.

Alternatively, $2^{125} \equiv 2^{119} \cdot 2^6 \equiv (2^7)^{17} \cdot 2^6 \equiv 1^{17} \cdot 2^6 \equiv 2^6 \equiv 64 \pmod{127}$.