

B.

We assume $S < A$ or $A < S$, so we can use one variable to calculate the time complexity
 Since base case $T(1)$ is a constant time:

$$\begin{aligned}
 T(n) &= T(n-1) + T(n-1) \\
 &= 2T(n-1) \\
 &= 2(2T(n-2)) \\
 &= 2(2(2T(n-3))) \\
 &= 2^k T(n-k) \\
 \text{let } n-k &= 1 \\
 k &= n-1 \\
 \therefore T(n) &= 2^{n-1} T(1) \\
 &\therefore O(2^n)
 \end{aligned}$$

C.

Time complexity: $O(n)$ (n is the length of the longest word).

Reason: "programming" VS. "pin"

1st call: compare "p" and "p"

2nd call: pass in (rogramming and in) compare "r" and "i"

3rd call: pass in (ogramming and in) compare "o" and "i"

.

.

.

9th call: pass in (ing and in) compare "i" and "i"

10th call: pass in (ng and n) compare "n" and "n"

11th call: string "pin" now is empty, return true