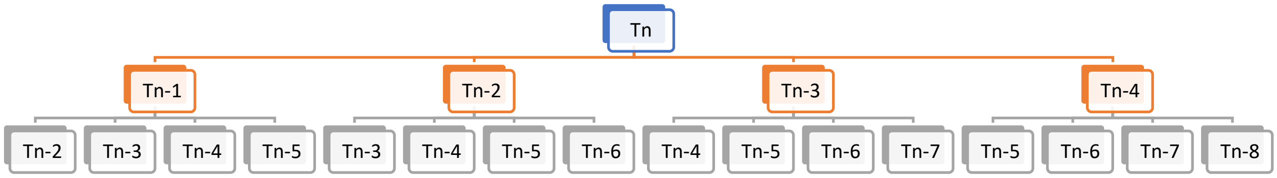
C)
For the first version:
Let T_n denotes the number of recursive calls by the ExponentialTetranacci(n) method.
T0 = 1;
T1 = 1;
T2 = 1;
T3 = 1;
T4 = 1;
T5 = T1+T2+T3+T4+1 = 1+1+1+1+1 = 5;
T6 = T5+T4+T3+T2+1 = 5+1+1+1+1 = 9;
T7 = T6+T5+T4+T3+1 = 9+5+1+1+1 = 17;
T8 = T7+T6+T5+T4+1 = 17+9+5+1+1 = 33;
T9 = T8+T7+T6+T5+1 = 33+17+9+5+1 = 64;
T10=T9+T8+T7+T6+1 = 64+33+17+9+1 = 124;
T11=T10+T9+T8+T7+1 = 124+64+33+17+1 = 239;
T12=T11+T10+T9+T8+1 = 239+124+64+33+1 = 461;
…
Tn = Tn-1 + Tn-2 + Tn-3 + Tn-4 + 1.

$Since\ T_{n-1} > T_{n-2} > T_{n-3} > T_{n-4},$
$T_n < 4T_{n-1} < 4^2T_{n-2} < 4^3T_{n-3} < \cdots < 4^{n-4}T_4 = 4^{n-4}$.
Therefore, we can conclude that Tn is $O(4^n)$.

for the second version:
T0 = 1;
T1 = 1;
T2 = 1;
T3 = 1;
T4 = 1;
T5 = T4+1 = 1+1 = 2;
T6 = T5+1 = 2+1 = 3;
T7 = T6+1 = 3+1 = 4;
T8 = T7+1 = 4+1 = 5;
T9 = T8+1 = 5+1 = 6;
T10 = T9+1 = 6+1 = 7;
T11 = T10+1 = 7+1 = 8;
…
Tn = Tn-1 + 1.
This means every recursive call decreases n by 1 when n > 4.
Therefore, $T_n \begin{cases} 1 & n \le 4 \\ n-3 & n > 4 \end{cases}$
we can conclude that Tn is O(n).

d)



When n>4, each recursive call will invoke 4 recursive calls, each of which will call another 4 recursive calls respectively; then, the program will execute till n=4. Therefore, the first version is exponential.