(1) $T(n) = \begin{cases} a & , n = 0 \\ b + nT(n-1), & n > 0 \end{cases}$

$$
\begin{aligned}
T(n) &= b + nT(n-1) \\
&= b + n(b + (n-1)T(n-2)) = b + nb + (n^2 - n)(T(n-2)) \\
&= b + nb + (n^2 - n)(b + (n-2)T(n-2)) \\
&= b(1 + n - n + n^2) + n \cdot (n-1)(n-2) \cdot T(n-3) \\
&= b(1 + n^{n-1}) + n! \, T(0) \\
&= b(1 + n^{n-1}) + a \cdot n!
\end{aligned}
$$

$T(n) = O(n!)$

(2) a) Algo 1

$T(n) = \begin{cases} 0 & , n = 1 \\ 1 & , n = 2 \\ n-1 & , n > 2 \end{cases}$

$T(n) = O(n)$

Algo 2

$T(n) = \begin{cases} 0 & , n = 0 \\ 1 + T(\lfloor n/2 \rfloor), & n > 0 \end{cases}$

$$
\begin{aligned}
T(n) &= 1 + T(\lfloor n/2 \rfloor) \\
&= 1 + (1 + T(\lfloor n/4 \rfloor)) = 2 + T(\lfloor n/4 \rfloor) \\
&= 3 + T(\lfloor n/8 \rfloor) \\
&\vdots \\
&= k + T(\lfloor n/2^k \rfloor)
\end{aligned}
$$

$n/2^k = 1 \longrightarrow \log(n/2^k) = \log 1$

$\log n - k \log 2 = 0$

$k = \dfrac{\log n}{\log 2} = {}^2\log n$

Sehingga

$$
\begin{aligned}
T(n) &= \lfloor {}^2\log n \rfloor + T(1) \\
&= \lfloor {}^2\log n \rfloor + n
\end{aligned}
$$

$T(n) = O({}^2\log n)$

b.) Algoritma kedua yang lebih mangkus karena

$O({}^2\log n) < O(n)$

GELATIK