1.
$f_n = f_0 + f_1x + f_2x^2 + f_3x^3 ... + f_nx^n$
$-8xf_n = 0 - 8f_0x - 8f_1x^2 - 8f_2x^3 - 8f_3x^4 ... - 8f_{n-1}x^n$
$-\sum_n 4^nx^n = -1 - 4x - 4^2x^2 - 4^3x^3 ... - 4^nx^n$
By adding them together:
$(1 - 8x)f_n = -\sum_n 4^nx^n$

$$f_n = \frac{1}{(1-4x)*(1-8x)} = \frac{c_1}{(1-4x)} + \frac{c_2}{(1-8x)}$$

$c_1 + c_2 = 1, 8c_1 + 4c_2 = 0$,solve them =¿ $c_1 = -1, c_2 = 2$

$$\frac{-1}{(1-4x)} + \frac{2}{(1-8x)}$$

$$\sum_n -1 * 4^nx^n + \sum_n 2 * 8^nx^n = \sum_n (-1 * 4^n + 2 * 8^n) * x^n$$

Thus $f_n = (-1 * 4^n + 2 * 8^n)$
2.
$p(n) = c * 4^n$
$c * 4^n = 7c * 4^{n-1} + \frac{18}{4} * 4^n$
$4^{n-1} * (-3c - 18) = 0$. Thus $c = -6$.The particular solution is $(-6) * 4^n$
We also find a solution to $g_n = c7^n$.The function should be $g_n = c7^n + (-6) * 4^n$

Now plug $g_0 = 1$,we get $1 = c - 6 => c = 7$,thus the solution to the original function is

$$g_n = 7^{n+1} + (-6) * 4^n$$

3. Divide and Conquer: Assume the initial size is n, and any further size becomes $n/(2^n)$,since it is a divide and conquer algorithm, its size decrease by half.We have formula

$$f_n = (-1 * 4^n + 2 * 8^n)$$

according to 1.Since $N = 2^n$,

$$f_n = (-1 * N^2 + 2 * N^3)$$

, as N can be as large as infinity,discarding all these constant and leave the term with highest power, we conclude this algorithm has a complexity of N^3.

Strassens Algorithm: According to 2.$g_n = 7^{n+1} + (-6) * 4^n$
$g_n = 7 * 7^n + (-6) * N^2$,and also we observe that $7^n = (2^n)^{log_2 7}$.Hence it is

$$g_n = 7 * N^{log_2 7} + (-6) * N^2$$

.By discarding constants and choosing the highest power term, we get the complexity for this one ,which is $N^{log_2 7}$.