There are 4 operations: +, -, * and @(concatenate).
1. dfs. $O(4^n)$.
2. meet in the middle. see http://maskray.me/blog/2015-10-16-leetcode-expression-add-operators.
1) if we can't cross the middle, when we merge the results, we need to merge (a, b) and (c, d) to get $a+bc+d$. let t be the target value, we want $a + bc + d = t$, i.e. the line $y = cx + d$ passes point $(b, t - a)$.
2) if we allow * and @ cross the middle, we only need to merge a and b to get $a + b$ or $a - b$. set $n' = \frac{2}{3}n$, the running time is $O(4^{n'} + 4^{n-n'} \cdot 2^{n'}) = O(2^{\frac{4}{3}n})$.
3) let the middle be $\frac{n}{2}$, assume we connect the letters in $[l, r]$ by $*$ or @ which crosses the middle, enumerate l and r s.t. $1 \le l \le \frac{n}{2} \le r \le n$. we can preprocess all results using +, -, *, @ (each is a single value) in $[1, l]$ and $[r, n]$ in $O(4^{\frac{n}{2}})$ (in the following we ignore $poly(n)$ factors). wlog assume $l \le n - r$, to combine the three parts $[1, l]$, $[l, r]$ and $[r, n]$, first fix the result in $[l, r]$ (there are $O(2^{n'})$ where $n' = r - l$), then enumrate the result in $[1, l]$, and use $O(1)$ to find the result in $[r, n]$. for any value of n', the running time is $O(2^{n'} \cdot 4^{\frac{n-n'}{2}}) = O(2^n)$. in total $\tilde{O}(2^n + k)$.

we need to solve the point-line incidence subproblem. suppose there are $m = 4^{n/2} = 2^n$ points and m lines, the number of point-line incidences is $O(m^{\frac{4}{3}})$ by Szemerédi-Trotter theorem [3] (and this is tight, see the construction in https://en.wikipedia.org/wiki/Szemer%C3%A9di%E2%80%93Trotter_theorem). a simplified proof was introduced in the book [2].
we can find all k point-line incidences in:
1) $O(m\sqrt{m}+k)$. divide the lines into groups of size x, for each group use $O(x^2)$ to compute the arrangement of lines, and $\tilde{O}(1)$ to locate each point in the arrangement. set $x = O(\sqrt{m})$, in total $\frac{m}{x} \cdot \tilde{O}(x^2+m) = \tilde{O}(m^{1.5})$.
2) let $\alpha = \log_2 \frac{1+\sqrt{5}}{2}$, $O(m^{\frac{2\alpha+1}{\alpha+1}} (\log m)^{\frac{1}{\alpha+1}} + k) \approx O(m^{1.41} + k)$ [1, Problem 14.6].

References

[1] Herbert Edelsbrunner. *Algorithms in combinatorial geometry*, volume 10. Springer Science & Business Media, 2012.

[2] Sariel Har-Peled. *Geometric approximation algorithms*. Number 173. American Mathematical Soc., 2011.

[3] Endre Szemerédi and William T. Trotter. Extremal problems in discrete geometry. *Combinatorica*, 3(3-4):381–392, 1983.