Letter combinations : Recursion

2 → a, b, c

3 → d, e, f

⋮

9 → w, x, y, z

given a string of digits, enumerate list of all possible strings it stands for

→ n digits ⇒ $3^n \le \# \le 4^n$

↑ no. of possible strings

→ will necc be exponential

Recursion : string S = s[0] + s[1:]

↗ {*, •, ✓, ?} ↗ answer = L

$L_{new} = [\ *y,\ \cdot y,\ \checkmark y,\ ?y \mid y \in L\ ]$

↑ return

base case: " " → [" "]

(slight adjustment : problem wants you to

return [ ] for " " — so have 2

base cases with a flag ← to detect if original input is " " or input " " is by a recursive call)

<u>complexity analysis</u>

n = length of input string

$f(n) = f(n-1) + O(n4^n)$

L = output of $n-1$ string, $|L| \leq 4^{n-1}$

$y \in L \Rightarrow$

$|y| = n-1$

concatenating $* + y$ ← $O(n)$

there are $4^{n-1} * 4$ concatenations

so 4^n concatenations, each is $O(n)$

so $O(n4^n)$

$f(n) = n4^n + (n-1)4^{n-1} + (n-2)4^{n-2} - - -$

$$1 + x + x^2 + \ldots + x^n = \frac{1-x^{n+1}}{1-x} = g(x)$$

$$+ \; --- (n-1)x^{n-1} + nx^n = g'(x) \cdot x$$

$$f(n) = g'(4) \cdot 4 = O(4^n)$$