Dynamic Programming

Note :- This playlist is only for explanation of Qns & solutions.

See my "DP Concepts & Qns" Playlist for understanding DP from Scratch...

codestorywithMIK

Facebook ⎤
Instagram ⎦→ codestorywithMIK

Twitter → cswithMIK

(WhatsApp) → codestorywithMIK

Leetcode
-1220
HARD

Count Vowels

Permutation

Company : ∞ Meta , Google

1220. Count Vowels Permutation

Hint ⋯

Hard 👍 2.7K 👎 183 ☆ ↻

🔒 Companies

$n = 3$

Given an integer n, your task is to count how many strings of length n can be formed under the following rules:

- Each character is a lower case vowel ('a', 'e', 'i', 'o', 'u')
- Each vowel 'a' may only be followed by an 'e'.
- Each vowel 'e' may only be followed by an 'a' or an 'i'.
- Each vowel 'i' **may not** be followed by another 'i'.
- Each vowel 'o' may only be followed by an 'i' or a 'u'.
- Each vowel 'u' may only be followed by an 'a'.

Since the answer may be too large, return it modulo 10^9 + 7.

$$e \longrightarrow a$$
$$\searrow i$$

$$i \rightarrow a$$
$$\searrow e$$
$$\searrow o$$
$$\searrow u$$
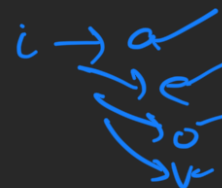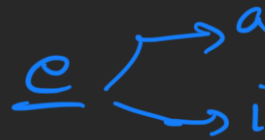
$$\frac{a}{} \bigg/ \frac{e}{} \bigg/ \frac{i}{} \bigg/ \frac{o}{} \bigg/ \frac{u}{}$$

Example 1:

$n = 1$

```
Input: n = 1
Output: 5
Explanation: All possible strings are: "a", "e", "i" , "o" and "u".
```

Example 2:

$$\underline{u} \quad \underline{a}$$

$$n = 2$$

```
Input: n = 2
Output: 10
Explanation: All possible strings are: "ae", "ea", "ei", "ia", "ie", "io", "iu",
"oi", "ou" and "ua".
```

```
Input: n = 5 ✓
Output: 68 ✓
```

How to think
from Scratch ...

Slidw: Tple Gk

(•) options   (Recursion)

$$\eta = 2$$

① $\dfrac{a}{0}$ $\quad\dfrac{e}{1}$ $(a,1) \rightarrow (e,0)$

② $\boxed{e}$ $\dfrac{a/i}{1}$ $(e,1) \rightarrow (a,0)$

     $(i,0)$

④ i $a/e/o/u$ $(a,0)$

②	$\dfrac{O}{0}$	$\dfrac{i/U}{}$
①	$\dfrac{U}{0}$	$\dfrac{a}{1}$

$(1,1) \rightarrow (e,0)\ 1.$
$\qquad\quad\rightarrow (0,0)\ 1.$
$\qquad\quad\rightarrow (U,0)\ 1.$

$(0,1) \rightarrow (i,0)\ 1$
$\qquad\quad\rightarrow (U,0)\ 1$

$\rightarrow (U,1) \rightarrow (a,0)\ 1$

10

$n = 2$

$O(n)$

Solve('a', n-1)
$\quad+$
Solve('e', n-1)
$\quad+$
Solve('i', n-1)
$\quad+$
Solve('0', n-1)
$\quad+$
Solve('U', n-1);

Solve (char ch, int n)

if (n == 0)
$\qquad$ return 1;
int result = 0
if (ch == 'a') {
$\quad$ result += solve('e', n-1);
} else if (ch == 'e') {

result += solve('a', n-1)
$\qquad\qquad + $ solve('i', n-1)
}
$\vdots$
i

$$O(u)$$
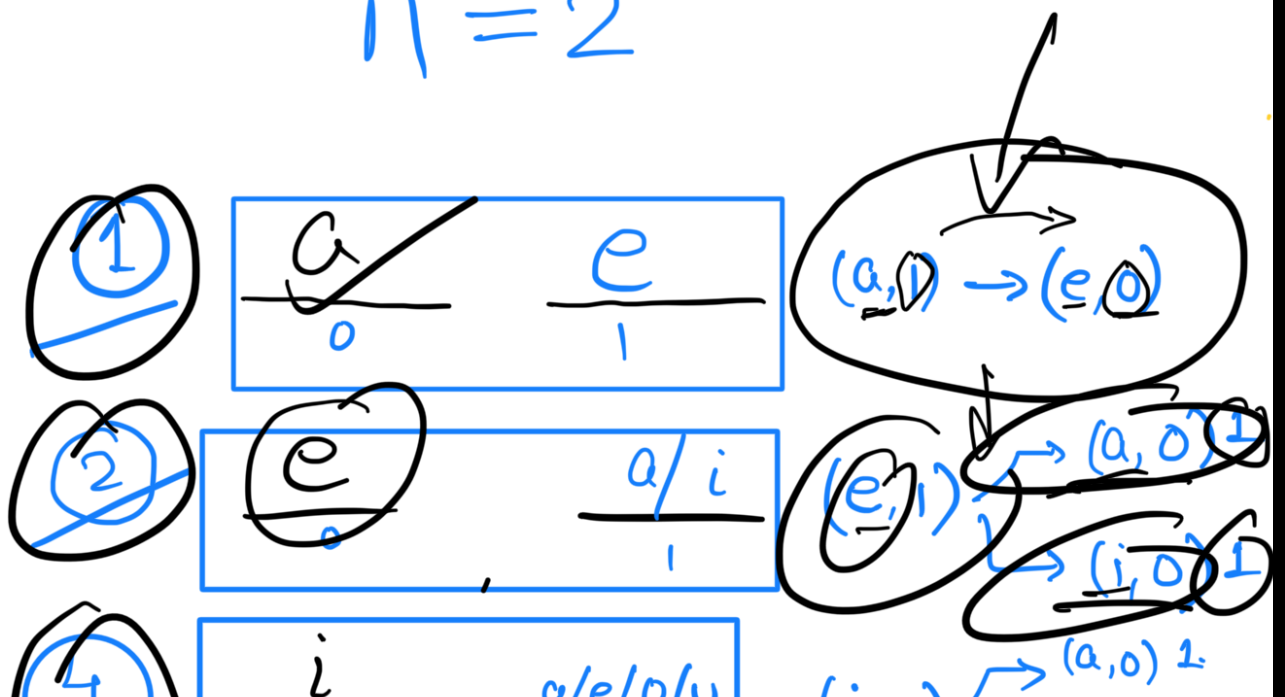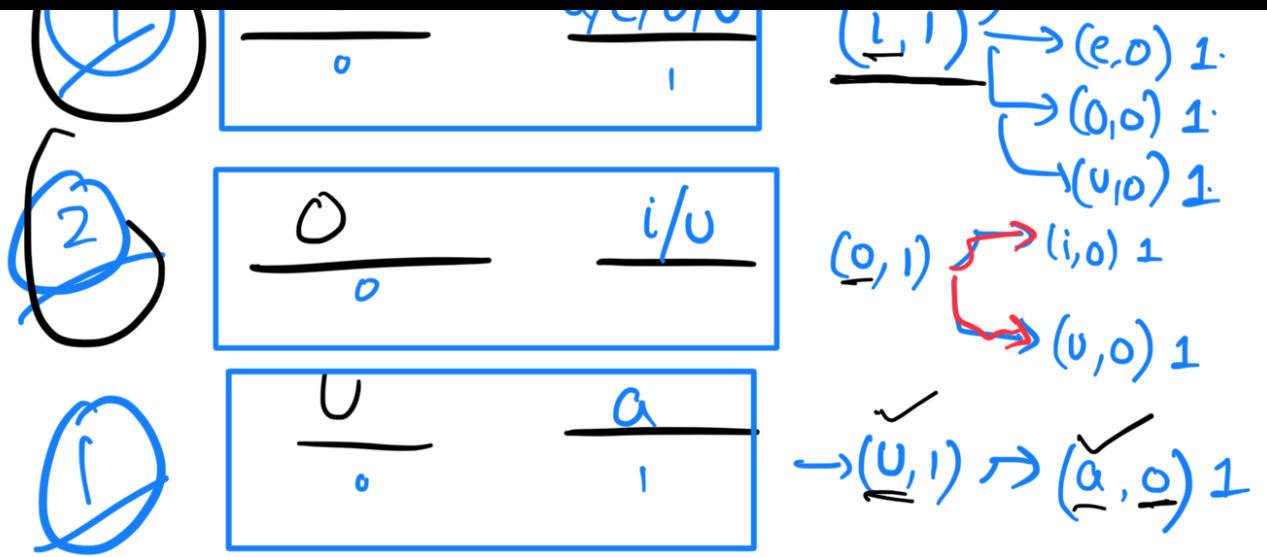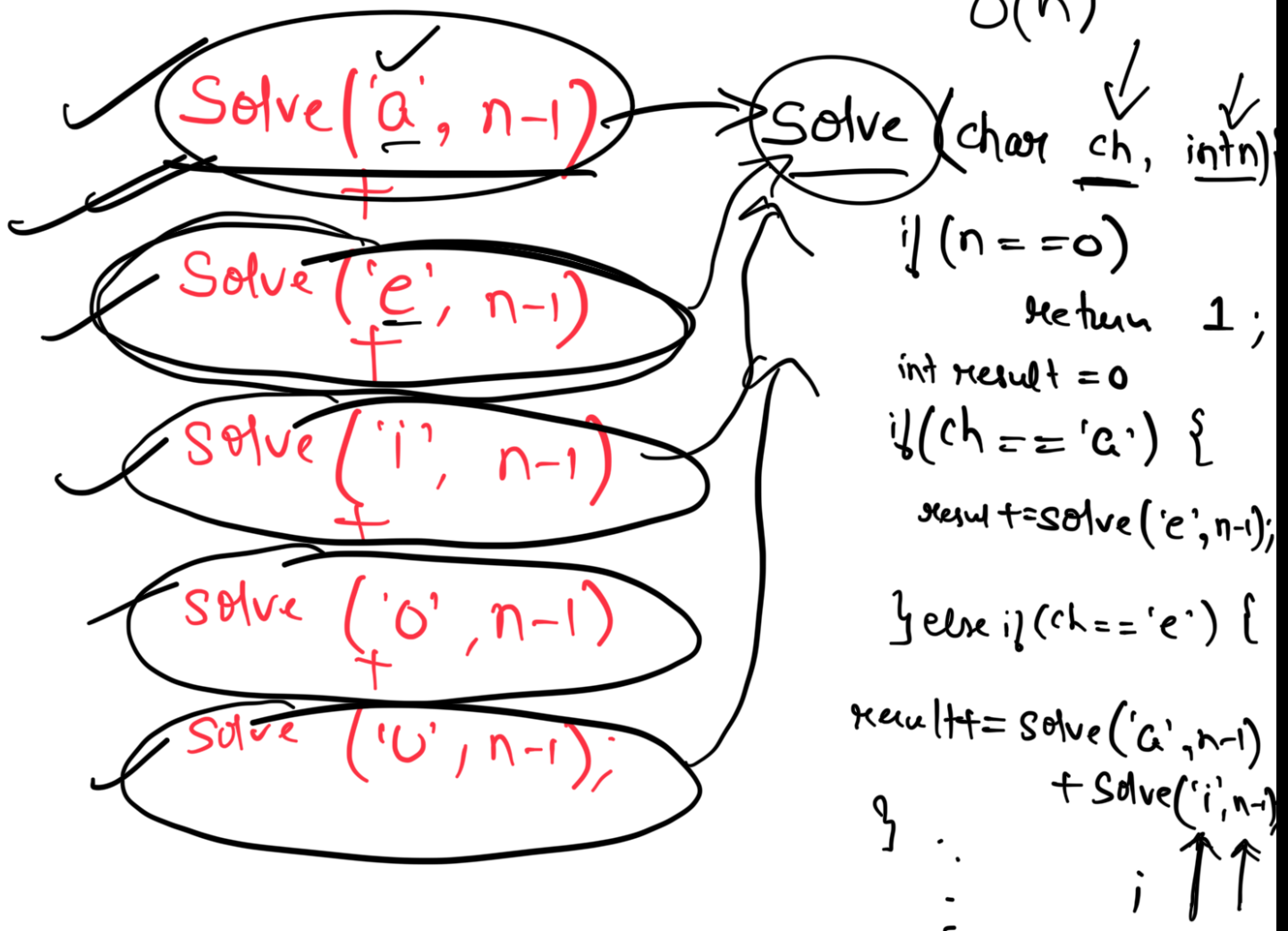
return result;
}

a, e, i, o, u

Memoize = $[ch][int]$

6      $n+1$

Bottom UP

Past

Smaller stats n

solve(ch, n)

a	"e"	"a"
i	e	a
e	i	a
o	i	a n=3
i	U	a
o	U	a
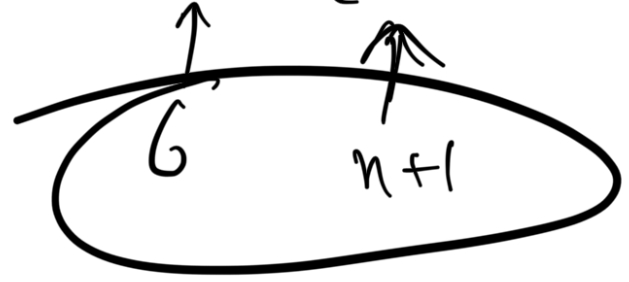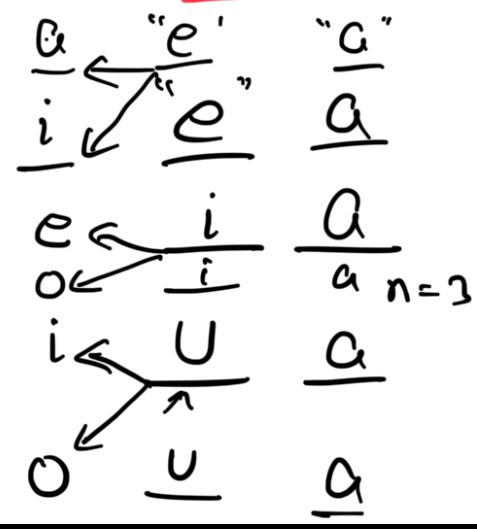
$a \to e$
$e \to a, i$
$i \to a, e, o, u$
$o \to i, u$
$u \to a$

$n = 3$
$\hookrightarrow n = 2$

State Definition :-

$$i=0 \longrightarrow a$$
$$i=1 \longrightarrow e$$
$$2 \longrightarrow i$$
$$3 \longrightarrow o$$

$$4 \longrightarrow 0$$

$$\boxed{f[i][j]} = \# \text{ of } \underset{\smile}{strings} \text{ ending } at$$

$$character \; i \qquad of \; \underline{length \; j}$$

$$f[0]$$
$$\downarrow$$

$$f[1]$$

$$f[2]$$

$$f[3]^{\swarrow}$$

$$f[4]^{\swarrow}$$

$$n-1 \qquad \overset{n-2}{} \qquad n$$

$$Solve \; f[a][n];$$
$$\underset{f}{}$$

$$f[e][n];$$
$$\underset{f}{}$$

$$\underline{\; i \;} \; f$$
$$\underline{\; 0 \;}$$
$$u$$
$$\underline{\; u \;}$$

Dry Run :-

$f[a][i] = 1$

$f[e][i] = 1 \rightarrow$ "e"

$f[i][i] = 1 \rightarrow$ "i"

$f[o][i] = 1$

$f[u][i] = 1$

leng = 2,
3
4.
... ~ n

$f[a][n] = x \Leftarrow$

$f[e][n] = y$

$f[i][n] = 2$

$f[o][n] = s$

$f[u][n] = k$ $f[5][n+1]$;

n

$O(n)$.

```
for ( int len = 2 ; len <= n ; len++){
    f[a][len] = f[e][len-1] + f[i][len-1]
                             + f[u][len-1];
    f[e][len] = f[a][...
```

$$= a[a]][len-1] + f[i][len-i]$$

$$f[i](len) \underline{\quad}$$

$$f[0](len) \underline{\quad}$$

$$f[0][len] \underline{\quad}$$

)