# 91. Decode Ways

'A' -> "1"

'B' -> "2"

...

"26" -> "26"

Example:

S = "12" output = 2

↓

"AB" or "L"

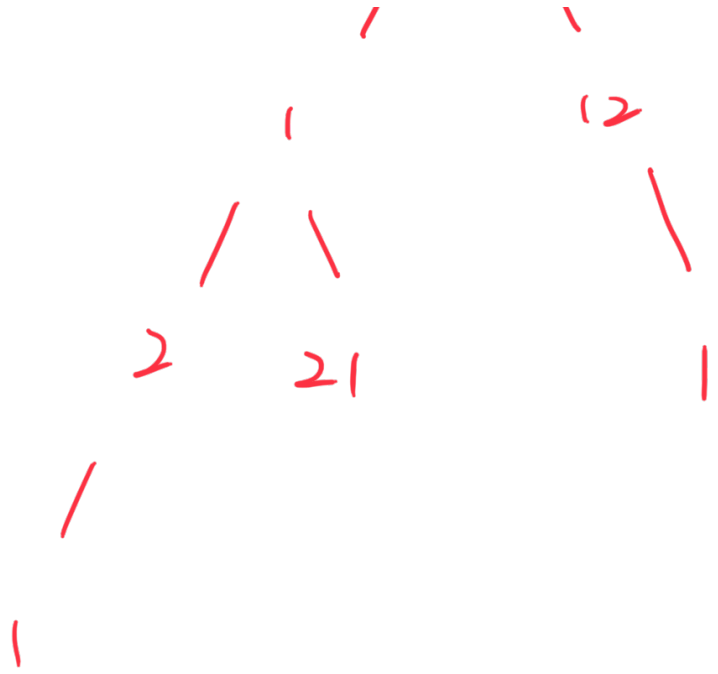
Analyse:

"121"

Backtracking.

Start

/ \




pseudo code:

```
def numDecoding:
    if len(S) == 0:
        return 0
    res = 0
    if len(S) >= 1 and S[0]:
        res += 1
    'res+= numDecoding(S[1:])
    if len(S) >= 2 & S[:2] <= 26
        res += 1 & S[0] != '0'
    res+= numDecoding(S[2:]
```

Works, but slow.

return res.

Generally speaking, it's okay but just too slow.

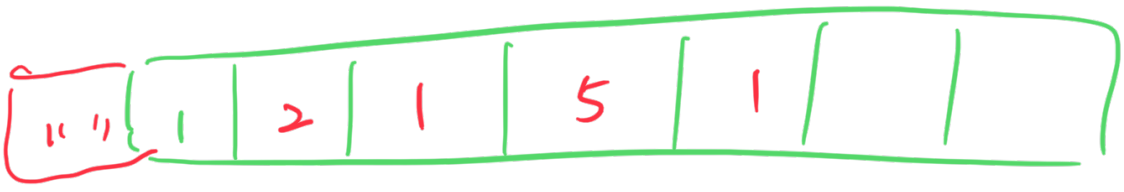
Dynamic programming:

$f(i) = f(i-1) + f(i-2)$

$f(i-1)$ → [1, 9]

$f(i-2)$ → [10, 26]

Analyse:



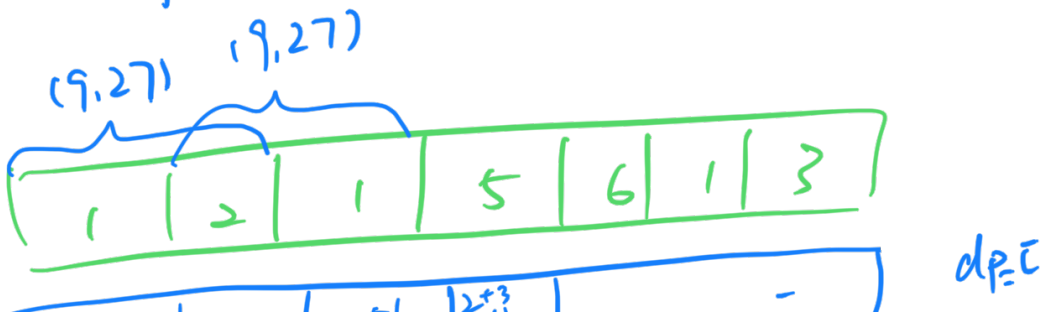

1 1 1+1 = 2 2+1 = 3 3+2 = 5 5+0 = 5

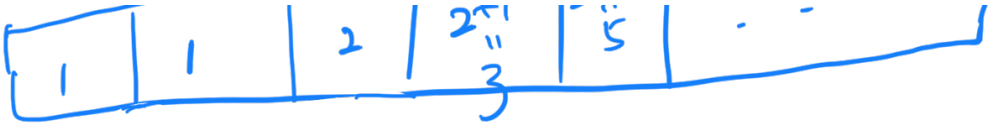
pseudo-code:

```
if (s = null || s.length == 0 ||
    s[0] == "0") : return 0
```

```
dp = [ len(s) + 1 ],
dp[0] = 1 , dp[1] = 1

for i in s.length:

    if s[i] != 0:

        dp[i+1] = dp[i]

    if 9 < s[i-1 : i+1] < 27:

        dp[i+1] += dp[i-1]:

return dp(s.len)
```

Let's follow the code:




1
1
2
3
5