Problem Definition:

給定一字典為 wordDict 和一字串為 s. 欲判斷 s 是否可利用字典中的字組成.

Example: ①.

s = "airplane"

wordDict = ["air", "plane"]

則為 true.

②.

s = "hahaha"

wordDict = ["ha"]

同樣為 true.

Solution:

① Dynamic Programming:

定義子問題為 $s[i:j]$ 是否可由 wordDict 組合而成.

設 optimal value 為 $d_{i,j}$ 則:

$$d_{i,j} = \begin{cases} \text{true} & \text{if } \exists\ k,\ i \le k \le j \text{ 且 } d_{i,k} = \text{true},\ d_{k+1,j} = \text{true} \\ \text{true} & \text{if } s[i:j] \text{ in wordDict} \\ \text{false} & \text{otherwise.} \end{cases}$$

For example: $s =$ "airplane"

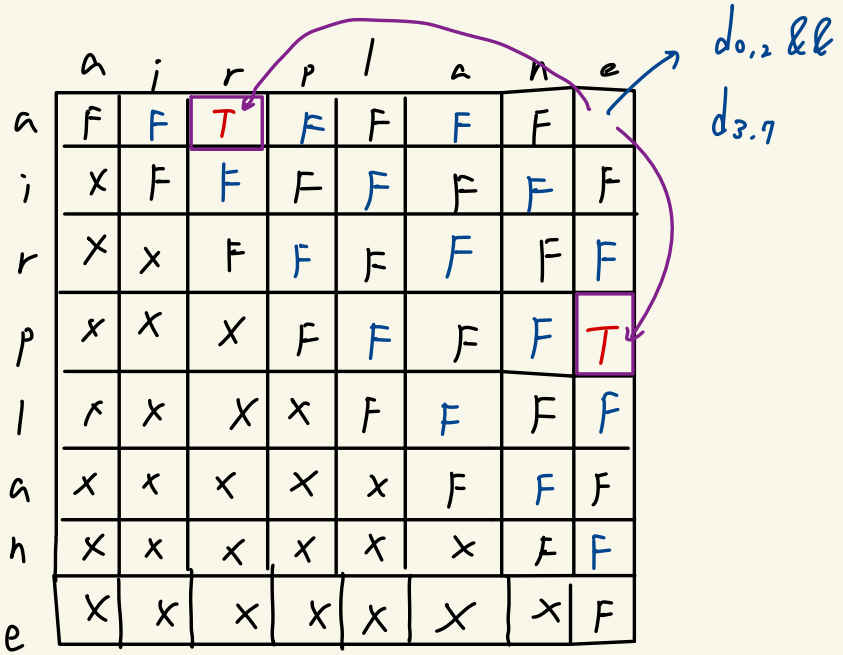
$d_{0,7} = \text{true}$ ∵ $\exists\ k = 2$, $d_{0,2} = \text{true}$ 且 $d_{3,7} = \text{true}$.

故計算：$d_{0,7}$ 時，需先算得：

$d_{0,1}$、$d_{2,7}$　$d_{0,2}$、$d_{3,7}$　$d_{0,3}$、$d_{4,7}$　$d_{0,4}$　$d_{5,7}$　$d_{0,5}$、$d_{6,7}$

故填表順序應為：$d_{0,1} \sim d_{1,2} \sim \cdots \sim d_{6,7}$

再來，才為：　$d_{0,2}$



②.

重新定義子問題為：

s[:i] 是否可由 wordDict 組合而成.

則 optimal value 為 d_i.

For example: s = "leetcode leet"

wordDict = ["leet", "code"]

則 d[3] = true. d[7] 為 true, d[11] 為 true.

則：

$$d[i] = \begin{cases} \text{true} & \text{if } \exists k < i \text{ 且 } d[k] \text{ 為 true 且 } s[k+1:i] \text{ in wordDict.} \\ \text{true} & \text{if } s[:i] \text{ in wordDict} \\ \text{false} & \text{otherwise} \end{cases}$$

共有 O(n) 个 state，每个 state 要遍历 0~i-1 且检查 substring s[k+1:i] 是否在 wordDict 中

Time: $O(n^3)$

事实上，因 wordDict 已知：

每个 state 中，可检查 wordDict 中每个 word，设 size 为 k

则：$d[i] = \begin{cases} \text{true} & \text{if } \exists \text{ - word 在 wordDict 中} \\ & \text{使得 } d[i-k] = \text{true 且} \\ & s[i-k+1:i] \text{ 为此 word.} \\ \text{true} & \text{if } \exists \text{ - word, } s[:i] \text{ 为此 word.} \end{cases}$

Ex:

s = "leetcodeleet"

wordDict = ["leet", "code"]

$d[7] = true$ ∵ 存在 "code"

+ $d[3] = true$

且 $s[4:7] =$ "code"

此时, Time Complexity 为: $O(n) \times O(m \cdot l)$

其中: m: wordDict size

l: average word len in wordDict.