

$s = \text{cars}$

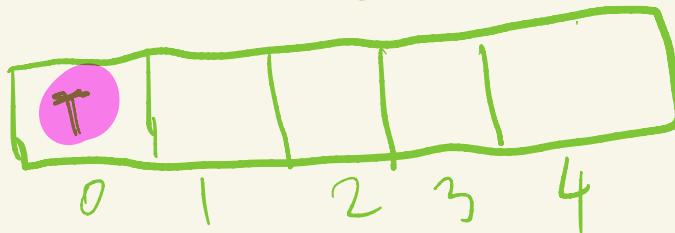
$\text{words} = [\text{car}, \text{a}, \text{rs}]$

we initialize arr. with

$\text{len}(s) + 1$

arr [$\text{len}(s) + 1$]

arr [4+1] \rightarrow arr [5]



every string contains empty string, so we assign

arr[0] = true

cars					
	T	F	T	T	T
0	1	2	3	4	

we should start

$i = 1$ and $j = 0 ; j < i ; j++$

$s[j:i] = s[0:1] = c$

c not exist in words

$\therefore s[1] = \text{False}$

$i = 2$ and $j = 0 ; j < i ; j++$

$s[2] = t$

$s[0:2] = ca \rightarrow (\text{True})$ if we get true we break for loop

$i = 3$

$s[0:3] = car \rightarrow T \therefore s[3] \rightarrow t$

$i = 4 ; j = 0 ; j < i ; j++$

$s[0:4] = \text{ars} \rightarrow F$

$i = 4 ; j = 1$

$s[1:4] = ars \rightarrow F$

$s[2:4] = rs \rightarrow T$

$a[4] = T$

case
= not valid

s = cats and dog

words = [cats, dog, sand, and, arr]

$$\text{arr}[\text{len}(s) + 1] = \text{arr}[10]$$

" " c a t s a n d o g

T	F	F	T	T	F	F	T	F	F
6	1	2	3	4	5	6	7	8	9

$$\text{arr}[0] = T$$

i = 1 ; j = 0 ; j < i ; j++

$s[0:1] = c \rightarrow F \rightarrow \text{arr}[1]$

$s[0:2] = ca \rightarrow F \rightarrow \text{arr}[2]$

$s[0:3] = cat \rightarrow F \rightarrow \text{arr}[3]$

$s[0:4] = cats \rightarrow F \rightarrow \text{arr}[4]$

$s[0:5] = \text{catsa} \rightarrow F$

$i=5, j=1$

$s[1:5] = \text{atsa} \rightarrow F$

$i=5, j=2$

$s[2:5] = \text{tsa} \rightarrow F$

$i=5, j=3$

$s[3:5] = \text{sa} \rightarrow F$

$i=5, j=4$

$s[4:5] = \text{a} \rightarrow F$

$\text{arr}[5] \rightarrow F$

$s[0:6] = \text{catsan} \rightarrow F$

$i=6, j=1 \rightarrow \text{catsan}$

$\text{arr}[j] \rightarrow \text{arr}[1] \rightarrow F$

$j=2; \text{arr}[2] = F (+\text{san})$

$j = 3$; $\text{arr}[3] = T$ but

$s[3:6] \rightarrow \text{tsa}$ not in word
 F

$j = 4 \rightarrow \text{arr}[4] = T$

$s[4:6] \rightarrow sa \rightarrow .F$

$T \& F = \text{F}$

$j = 5 \rightarrow \text{arr}[5] = F$

$\text{F} \rightarrow \text{arr}[6]$

$i = 7,$

$s[0:7] = \boxed{\text{atsand}} \rightarrow \text{F}$

$s[1:7] = \text{atsand} \rightarrow F$

$s[2:7] = \text{tsand} \rightarrow F$

$s[3:7] = \text{sand} \rightarrow \text{F} / \text{arr}[7]$

i = 8

$s[\text{0}; 8] = \text{catsand} \circ \rightarrow^F$

`ss[1:2] = afsano` → F

$s[2:7] = "sand" \rightarrow F$

$S[2:8] = \text{sand} \rightarrow F$

$S[4:8] = \text{and} \rightarrow P$

$$S[S:8] = n_{dp} > \infty$$

$$\cup \{f: \exists\} = d_0 \rightarrow F$$

$S[7:8] = 0 \rightarrow \text{F} - \text{err}[8]$

$i - a$

$$S[0:9] = \boxed{\text{Crafts andology}}$$

hrr[9]

```
func wordBreak(s string, wordDict []string) bool {
    dp := make([]bool, len(s)+1)
    dp[0] = true

    mapWordDict := make(map[string]bool)
    for _, word := range wordDict {
        mapWordDict[word] = true
    }

    for i := 1; i <= len(s); i++ {
        for j := 0; j < i; j++ {
            substr := s[j:i]
            if dp[j] && mapWordDict[substr] {
                dp[i] = true
                break
            }
        }
    }

    return dp[len(dp)-1]
```

check $[0:j]$ is valid ?

check substr
contain in
word