Problem # 290 : Word Pattern

<https://leetcode.com/problems/word-pattern/>

Solution:

[**https://leetcode.com/problems/word-pattern/discuss/834929/Simple-Python-3-solution-with-dictionaries.-Runtime-beats-82**](https://leetcode.com/problems/word-pattern/discuss/834929/Simple-Python-3-solution-with-dictionaries.-Runtime-beats-82)

Simple Python 3 solution with dictionaries. -- Runtime beats 82%

1. If length of string.split() and length of pattern are not equal, then return False
2. Create a dictionary called pat\_dict which checks the one-to-one mapping (or correspondence) from pattern to string.
3. Create another dictionary called str\_dict which checks the one-to-one mapping(or correspondence) from string to pattern.
4. If all above conditions are met, return True

Class Solution:

def wordPattern(self, pattern: str, str: str) -> bool:

str\_list = str.split()

if len(str\_list) != len(pattern):

return (False)

pat\_dict = {} # initialize. dictionary -- key = pattern, value = str

str\_dict = {} # initialize. dictionary -- key = str, value = pattern

for i, word in enumerate(str\_list):

if pattern[i] not in pat\_dict.keys():

pat\_dict[pattern[i]] = str\_list[i]

else:

if pat\_dict[pattern[i]] != str\_list[i]:

return (False)

if str\_list[i] not in str\_dict.keys():

str\_dict[str\_list[i]] = pattern[i]

else:

if str\_dict[str\_list[i]] != pattern[i]:

return (False)

return (True)