Description | Editorial | Solutions (5.1K) | Submissions

139. Word Break

Medium | 15.5K | 649

Companies

Given a string `s` and a dictionary of strings `wordDict`, return `true` if `s` can be segmented into a space-separated sequence of one or more dictionary words.

Note that the same word in the dictionary may be reused multiple times in the segmentation.

Example 1:

```
Input: s = "leetcode", wordDict = ["leet","code"]
Output: true
Explanation: Return true because "leetcode" can be segmented as "leet code".
```

Example 2:

```
Input: s = "applepenapple", wordDict = ["apple","pen"]
Output: true
Explanation: Return true because "applepenapple" can be segmented as "apple pen apple".
Note that you are allowed to reuse a dictionary word.
```

Example 3:

```
Input: s = "catsandog", wordDict =
["cats","dog","sand","and","cat"]
```

Java | Auto

```java
class Solution {
    public boolean wordBreak(String s, List<String> wordDict) {
        HashMap<String,Integer>h1=new HashMap<>();
        for(String str:wordDict){
            h1.put(str,1);
        }
        int n=s.length();
        int dp[]=new int[n+1];
        Arrays.fill(dp,-1);
        return helper(s,0,h1,dp)==1;
    }
    int  helper(String s,int i,HashMap<String,Integer>h1,int dp[]){
        if(i==s.length()){
            return 1;
        }
        if(dp[i]!=-1){
            return dp[i];
        }
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

s =
"leetcode"

wordDict =
["leet","code"]

Console | Run | Submit

Description | Editorial | Solutions (7.5K) | Submissions

69. Sqrt(x)

Hint

Easy 7K 4.2K

Companies

Given a non-negative integer `x`, return *the square root of* `x` *rounded down to the nearest integer*. The returned integer should be **non-negative** as well.

You **must not use** any built-in exponent function or operator.

- For example, do not use `pow(x, 0.5)` in c++ or `x ** 0.5` in python.

Example 1:

```
Input: x = 4
Output: 2
Explanation: The square root of 4 is 2, so we return 2.
```

Example 2:

```
Input: x = 8
Output: 2
Explanation: The square root of 8 is 2.82842..., and since
we round it down to the nearest integer, 2 is returned.
```

Constraints:

- $0 <= x <= 2^{31} - 1$

Java | Auto

```java
class Solution {
    public int mySqrt(int x) {

        if (x == 0 || x == 1)
            return x;
        int start = 1;
        int end = x;
        int mid = -1;


        while (start <= end) {

            mid = start + (end - start) / 2;


            if ((long) mid * mid > (long) x)
                end = mid - 1;
            else if (mid * mid == x)
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

x =
4

Output

2

Console | Run | Submit