

# 0127. 单词接龙

👤 [ITCharge](#) ⌚ 大约 1 分钟

---

- 标签：广度优先搜索、哈希表、字符串
- 难度：困难

## 题目链接

---

- [0127. 单词接龙 - 力扣](#)

## 题目大意

---

给定两个单词 `beginWord` 和 `endWord`，以及一个字典 `wordList`。找到从 `beginWord` 到 `endWord` 的最短转换序列中的单词数目。如果不存在这样的转换序列，则返回 0。

转换需要遵守的规则如下：

- 每次转换只能改变一个字母。
- 转换过程中的中间单词必须为字典 `wordList` 中的单词。

## 解题思路

---

广度优先搜索。使用队列存储将要遍历的单词和单词数目。

从 `beginWord` 开始变换，把单词的每个字母都用 `a ~ z` 变换一次，变换后的单词是否是 `endWord`，如果是则直接返回。

否则查找变换后的词是否在 `wordList` 中。如果在 `wordList` 中找到就加入队列，找不到就输出 0。然后按照广度优先搜索的算法需要遍历队列中的节点，直到所有单词都出队时结束。

# 代码

py

```
class Solution:
    def ladderLength(self, beginWord: str, endWord: str, wordList: List[str]) ->
int:
    if not wordList or endWord not in wordList:
        return 0
    word_set = set(wordList)
    if beginWord in word_set:
        word_set.remove(beginWord)

    queue = collections.deque()
    queue.append((beginWord, 1))
    while queue:
        word, level = queue.popleft()
        if word == endWord:
            return level

        for i in range(len(w    )):
            for j in range(2, .
                new_word = word[:i] + chr(ord('a') + j) + word[i + 1:]
                if new_word in word_set:
                    word_set.remove(new_word)
                    queue.append((new_word, level + 1))

    return 0
```