# Problem 3016: Minimum Number of Pushes to Type Word II

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 79.93%
**Paid Only:** No
**Tags:** Hash Table, String, Greedy, Sorting, Counting

## Problem Description

You are given a string `word` containing lowercase English letters.

Telephone keypads have keys mapped with **distinct** collections of lowercase English letters, which can be used to form words by pushing them. For example, the key `2` is mapped with `["a","b","c"]`, we need to push the key one time to type `"a"`, two times to type `"b"`, and three times to type `"c"` _._

It is allowed to remap the keys numbered `2` to `9` to **distinct** collections of letters. The keys can be remapped to **any** amount of letters, but each letter **must** be mapped to **exactly** one key. You need to find the **minimum** number of times the keys will be pushed to type the string `word`.

Return _the**minimum** number of pushes needed to type _`word` _after remapping the keys_.

An example mapping of letters to keys on a telephone keypad is given below. Note that `1`, `*`, `#`, and `0` do **not** map to any letters.

![](https://assets.leetcode.com/uploads/2023/12/26/keypaddesc.png)

**Example 1:**

![](https://assets.leetcode.com/uploads/2023/12/26/keypadv1e1.png)

**Input:** word = "abcde" **Output:** 5 **Explanation:** The remapped keypad given in the image provides the minimum cost. "a" -> one push on key 2 "b" -> one push on key 3 "c" -> one push on key 4 "d" -> one push on key 5 "e" -> one push on key 6 Total cost is 1 + 1 + 1 + 1 + 1 = 5. It can be shown that no other mapping can provide a lower cost.

**Example 2:**

![](https://assets.leetcode.com/uploads/2024/08/20/edited.png)

**Input:** word = "xyzxyzxyzxyz" **Output:** 12 **Explanation:** The remapped keypad given in the image provides the minimum cost. "x" -> one push on key 2 "y" -> one push on key 3 "z" -> one push on key 4 Total cost is 1 * 4 + 1 * 4 + 1 * 4 = 12 It can be shown that no other mapping can provide a lower cost. Note that the key 9 is not mapped to any letter: it is not necessary to map letters to every key, but to map all the letters.

**Example 3:**

![](https://assets.leetcode.com/uploads/2023/12/27/keypadv2.png)

**Input:** word = "aabbccddeeffgghhiiiiii" **Output:** 24 **Explanation:** The remapped keypad given in the image provides the minimum cost. "a" -> one push on key 2 "b" -> one push on key 3 "c" -> one push on key 4 "d" -> one push on key 5 "e" -> one push on key 6 "f" -> one push on key 7 "g" -> one push on key 8 "h" -> two pushes on key 9 "i" -> one push on key 9 Total cost is 1 * 2 + 1 * 2 + 1 * 2 + 1 * 2 + 1 * 2 + 1 * 2 + 1 * 2 + 2 * 2 + 6 * 1 = 24. It can be shown that no other mapping can provide a lower cost.

**Constraints:**

* `1 <= word.length <= 105` * `word` consists of lowercase English letters.

## Code Snippets

**C++:**

```
class Solution {
public:
int minimumPushes(string word) {

}
};
```

### Java:

```
class Solution {
public int minimumPushes(String word) {

}
}
```

### Python3:

```
class Solution:
def minimumPushes(self, word: str) -> int:
```