

Problem 3253: Construct String with Minimum Cost (Easy)

Problem Information

Difficulty: **Medium**

Acceptance Rate: 58.93%

Paid Only: Yes

Problem Description

You are given a string `target`, an array of strings `words`, and an integer array `costs`, both arrays of the same length.

Imagine an empty string `s`.

You can perform the following operation any number of times (including **zero**):

* Choose an index `i` in the range `[0, words.length - 1]`. * Append `words[i]` to `s`. * The cost of operation is `costs[i]`.

Return the **minimum** cost to make `s` equal to `target`. If it's not possible, return -1.

Example 1.

Input: `target = "abcdef"`, `words = ["abdef", "abc", "d", "def", "ef"]`, `costs = [100, 1, 1, 10, 5]`

Output: 7

Explanation.

The minimum cost can be achieved by performing the following operations:

* Select index 1 and append `"abc"` to `s` at a cost of 1, resulting in `s = "abc"`. * Select index 2 and append `"d"` to `s` at a cost of 1, resulting in `s = "abcd"`. * Select index 4 and append `"ef"` to `s` at a cost of 5, resulting in `s = "abcdef"`.

****Example 2:****

****Input:**** target = "aaaa", words = ["z","zz","zzz"], costs = [1,10,100]

****Output:**** -1

****Explanation:****

It is impossible to make `s` equal to `target`, so we return -1.

****Constraints:****

* `1 <= target.length <= 2000` * `1 <= words.length == costs.length <= 50` * `1 <= words[i].length <= target.length` * `target` and `words[i]` consist only of lowercase English letters. * `1 <= costs[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
    int minimumCost(string target, vector<string>& words, vector<int>& costs) {

    }
};
```

Java:

```
class Solution {
    public int minimumCost(String target, String[] words, int[] costs) {

    }
}
```

Python3:

```
class Solution:
    def minimumCost(self, target: str, words: List[str], costs: List[int]) ->
    int:
```

