

Problem 433: Minimum Genetic Mutation

Problem Information

Difficulty: Medium

Acceptance Rate: 56.10%

Paid Only: No

Tags: Hash Table, String, Breadth-First Search

Problem Description

A gene string can be represented by an 8-character long string, with choices from `'A'`, `'C'`, `'G'`, and `'T'`.

Suppose we need to investigate a mutation from a gene string `startGene`` to a gene string `endGene`` where one mutation is defined as one single character changed in the gene string.

* For example, `"AACCGGTT" --> "AACCGGTA"` is one mutation.

There is also a gene bank `bank`` that records all the valid gene mutations. A gene must be in `bank`` to make it a valid gene string.

Given the two gene strings `startGene`` and `endGene`` and the gene bank `bank``, return `_the minimum number of mutations needed to mutate from_`startGene` _to_`endGene``. If there is no such a mutation, return `-1``.

Note that the starting point is assumed to be valid, so it might not be included in the bank.

Example 1:

Input: `startGene = "AACCGGTT", endGene = "AACCGGTA", bank = ["AACCGGTA"]`

Output: `1`

Example 2:

Input: `startGene = "AACCGGTT", endGene = "AAACGGTA", bank = ["AACCGGTA", "AACCGCTA", "AAACGGTA"]`

Output: `2`

****Constraints:****

* `0 <= bank.length <= 10` * `startGene.length == endGene.length == bank[i].length == 8` *
`startGene`, `endGene`, and `bank[i]` consist of only the characters `['A', 'C', 'G', 'T']`.

Code Snippets

C++:

```
class Solution {
public:
    int minMutation(string startGene, string endGene, vector<string>& bank) {

    }
};
```

Java:

```
class Solution {
    public int minMutation(String startGene, String endGene, String[] bank) {

    }
}
```

Python3:

```
class Solution:
    def minMutation(self, startGene: str, endGene: str, bank: List[str]) -> int:
```