# 题目

稀疏数组搜索。有个排好序的字符串数组，其中散布着一些空字符串，编写一种方法，找出给定字符串的位置。

示例1:

输入: words = ["at", "", "", "", "ball", "", "", "car", "", "","dad", "", ""], s = "ta"

输出：-1

说明: 不存在返回-1。

示例2:

输入：words = ["at", "", "", "", "ball", "", "", "car", "", "","dad", "", ""], s = "ball"

输出：4

提示:

words的长度在[1, 1000000]之间

# 分析

## 方法一：二分查找

思路：

这个问题可以使用二分查找来解决。由于数组中可能存在空字符串，因此在查找时需要做一些调整：

1、如果中间位置的字符串为空字符串，则向左右两侧分别查找最近的非空字符串，并将中间位置更新为该位置。

2、根据更新后的中间位置与目标字符串的比较结果，决定向左或向右继续二分查找。

代码：

class Solution {

public:

int findString(vector<string>& words, string s) {

int left = 0, right = words.size() - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

int original\_mid = mid;

// 处理中间位置为""的情况

while (mid < right && words[mid] == "") {

mid++;

}

if (words[mid] == "") {

right = original\_mid - 1;

continue;

}

// 正常二分查找

if (words[mid] == s) {

return mid;

} else if (words[mid] < s) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

};