算法设计与应用基础：作业 2

杨博，林小拉

2020 年 6 月 15 日

DUE：2020/6/30

提交说明

- 请将作业以 **PDF** 附件形式发送到邮箱：algo2020@163.com
- 邮件主题及作业文件统一命名：第几次作业 _ 学号 _ 姓名，如，2_18XXXXXX_ 张
- 编程题一般是 OJ 平台 LeetCode 上的题目，点击题名即可跳转到题目对应的页面。对于编程题，要求在作业中写出四项内容：算法思路，复杂度分析，代码和 Accepted 截图。

作业

1. Number of Islands

   Given a $2d$ grid map of 1s (land) and 0s (water), count the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

   Example 1:

   Input :

$$\begin{matrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

   Output : 1

   Example 2:

   Input :

$$\begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{matrix}$$

   Output : 3

2. Word Ladder

   Given two words (*beginWord* and *endWord*), and a dictionary's word list, find the length of shortest transformation sequence from *beginWord* to *endWord*, such that:

   (a) Only one letter can be changed at a time.
   (b) Each transformed word must exist in the word list.

   Note:

   - Return 0 if there is no such transformation sequence.
   - All words have the same length.
   - All words contain only lowercase alphabetic characters.
   - You may assume no duplicates in the word list.
   - You may assume *beginWord* and *beginWord* are non-empty and are not the same.

   Example 1:

   Input :

   $beginWord = "hit",$
   $endWord = "cog",$
   $wordList = ["hot", "dot", "dog", "lot", "log", "cog"]$

   Output : 5

   Explanation: As one shortest transformation is $"hit" - > "hot" - > "dot" - > "dog" - > "cog"$, return its length 5.

   Example 2:

   Input :

   $beginWord = "hit",$
   $endWord = "cog",$
   $wordList = ["hot", "dot", "dog", "lot", "log"]$

   Output : 0

   Explanation: The *endWord* *"cog"* is not in *wordList*, therefore no possible transformation.

3. Shortest Path Visiting All Nodes

   An undirected, connected graph of N nodes (labeled $0, 1, 2, ..., N-1$) is given as graph.

   $graph.length = N$, and $j! = i$ is in the list $graph[i]$ exactly once, if and only if nodes i and j are connected.

   Return the length of the shortest path that visits every node. You may start and stop at any node, you may revisit nodes multiple times, and you may reuse edges.

   Example 1:

   Input : $[[1, 2, 3], [0], [0], [0]]$

   Output : 4

Explanation: One possible path is $[1, 0, 2, 0, 3]$

Example 2:

Input : $[[1], [0, 2, 4], [1, 3, 4], [2], [1, 2]]$

Output : 4

Explanation: One possible path is $[0, 1, 4, 2, 3]$

4. Jump Game

   Given an array of non-negative integers, you are initially positioned at the first index of the array.

   Each element in the array represents your maximum jump length at that position.

   Determine if you are able to reach the last index.

   Example 1:

   Input : $nums = [2, 3, 1, 1, 4]$

   Output : $true$

   Explanation: Jump 1 step from index 0 to 1, then 3 steps to the last index.

   Example 2:

   Input : $nums = [3, 2, 1, 0, 4]$

   Output : $false$

   Explanation: You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.

5. Gas Station

   There are N gas stations along a circular route, where the amount of gas at station i is $gas[i]$.

   You have a car with an unlimited gas tank and it costs $cost[i]$ of gas to travel from station i to its next station $(i + 1)$. You begin the journey with an empty tank at one of the gas stations.

   Return the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise return -1.

   Note:

   - If there exists a solution, it is guaranteed to be unique.
   - Both input arrays are non-empty and have the same length.
   - Each element in the input arrays is a non-negative integer.

   Example 1:

   Input:

   $gas = [1, 2, 3, 4, 5]$
   $cost = [3, 4, 5, 1, 2]$

Output: 3

Explanation:

Start at station 3 (index 3) and fill up with 4 unit of gas. Your tank $= 0 + 4 = 4$ Travel to station 4. Your tank $= 4 - 1 + 5 = 8$ Travel to station 0. Your tank $= 8 - 2 + 1 = 7$ Travel to station 1. Your tank $= 7 - 3 + 2 = 6$ Travel to station 2. Your tank $= 6 - 4 + 3 = 5$ Travel to station 3. The cost is 5. Your gas is just enough to travel back to station 3. Therefore, return 3 as the starting index.

Example 2:

Input:

$gas = [2, 3, 4]$
$cost = [3, 4, 3]$

Output: -1

Explanation:

You can't start at station 0 or 1, as there is not enough gas to travel to the next station. Let's start at station 2 and fill up with 4 unit of gas. Your tank $= 0 + 4 = 4$ Travel to station 0. Your tank $= 4 - 3 + 2 = 3$ Travel to station 1. Your tank $= 3 - 3 + 3 = 3$ You cannot travel back to station 2, as it requires 4 unit of gas but you only have 3. Therefore, you can't travel around the circuit once no matter where you start.