

# 12th May 2023 Session 11B

## Exercise 1. Number of islands

Given an  $m \times n$  2D binary `grid` which represents a map of "1" s (land) and "0" s (water), return 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.

```
def numIslands(self, G: List[List[str]]) -> int:  
    ...
```

Example 1:

```
Input: grid = [  
    ["1","1","1","1","0"],  
    ["1","1","0","1","0"],  
    ["1","1","0","0","0"],  
    ["0","0","0","0","0"]  
]
```

Output: 1

Example 2:

```
Input: grid = [  
    ["1","1","0","0","0"],  
    ["1","1","0","0","0"],  
    ["0","0","1","0","0"],  
    ["0","0","0","1","1"]  
]
```

Output: 3

## Exercise 2. Word Distance

A *transformation* sequence from word `beginWord` to word `endWord` using a dictionary `wordList` is a sequence of words  $beginWord \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_k$  such that:

- Every adjacent pair of words differs by a single letter
- Every  $s_i$  for  $1 \leq i \leq k$  is in `wordList`. Note that `beginWord` does not need to be in `wordList`
- $s_k == endWord$

Given two words, `beginWord` and `endWord`, and a dictionary `wordList`, return *the number of words* in the *shortest transformation* sequence from `beginWord` to `endWord`, or 0 if no such sequence exists.

```
def wordDistance(self, beginWord: str, endWord: str, wordList: List[str]) -> int:
```

Input: `beginWord = "hit", endWord = "cog", wordList = ["hot","dot","dog","lot","log","cog"]`

Output: 5

Explanation: One shortest transformation sequence is

"hit" -> "hot" -> "dot" -> "dog" -> "cog", which is 5 words long.

Input: `beginWord = "hit", endWord = "cog", wordList = ["hot","dot","dog","lot","log"]`

Output: 0

Explanation: The endWord "cog" is not in wordList, therefore there is no valid transformation sequence.