CSE 374: Algorithms I (Spring 2020)

Coding Homework #4 BFS/DFS

Grading Rubric:

1. The program submitted for this homework must pass the necessary base case test(s) in order to qualify for earning any score at all. Programs that do not meet base case requirements will be assigned zero!
2. The code should be well formatted and commented on. The basic requirement is the code is understandable for a person who has basic coding knowledge.
3. There is some additional test case will be used for grading. Your code must complete it correctly and efficiently

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

1. Only one letter can be changed at a time
2. Each transformed word must exist in the word list. Note that beginWord is not a transformed word.

Note:

1. Return an empty list if there is no such transformation sequence.
2. All words have the same length.
3. All words contain only lowercase alphabetic characters.
4. You may assume no duplicates in the word list.
5. You may assume beginWord and endWord are non-empty and are not the same.

Example 1:

Input:

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

Output:

```
[
  ["hit","hot","dot","dog","cog"],
  ["hit","hot","lot","log","cog"]
]
```

Example 2:

Input:

```
beginWord = "hit"
```

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

Output:
[]

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