127. Word Ladder

QuestionEditorial Solution
 My Submissions

- Total Accepted: **81072**
- Total Submissions: **411224**
- Difficulty: **Medium**

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:

1. Only one letter can be changed at a time
2. Each intermediate word must exist in the word list

For example,

Given:

beginWord = "hit"

endWord = "cog"

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

As one shortest transformation is "hit" -> "hot" -> "dot" -> "dog" -> "cog",

return its length 5.

```
class Solution {
public:
    int ladderLength(string beginWord, string endWord, unordered_set<string>& wordList)
{
        int bword_len = beginWord.length();
        int eword_len = endWord.length();

        if (bword_len == 0 || eword_len == 0 || bword_len != eword_len)
            return 0;

        queue<string> word_path;
        int trans_len = 0;
        int count = 0;

        word_path.push(beginWord);
        trans_len = 1;

        while (!word_path.empty())
        {
            count = word_path.size();

            while (count--)/* in the same level  */
            {
                string word = word_path.front();
                word_path.pop();
                int len = word.size();
                for (int i = 0; i < len; i++)
                {
                    char ch = word[i];

                    for (char k = 'a'; k <= 'z'; k++)
                    {
                        if (ch == k)
                            continue;

                        word[i] = k;

                        if (word == endWord)
                            return trans_len + 1;

                        if (wordList.find(word) != wordList.end())
                        {
                            word_path.push(word);
                            wordList.erase(word);
                        }
                    }

                    word[i] = ch;
                }
            }
            if (!word_path.empty())
                trans_len++;
        }

        return 0;
    }
};
```