import java.util.LinkedList;

import java.util.Set;

class Word{

String word;

int numSteps;

public Word(String word, int numSteps){

this.word = word;

this.numSteps = numSteps;

}

}

public class WordLadder {

public int ladderLength(String beginWord, String endWord, Set<String> wordDict) {

LinkedList<Word> queue = new LinkedList<Word>();

queue.add(new Word(beginWord, 1));

wordDict.add(endWord);

while(!queue.isEmpty()){

Word top = queue.remove();

String word = top.word;

if(word.equals(endWord)){

return top.numSteps;

}

char[] arr = word.toCharArray();

for(int i=0; i<arr.length; i++){

for(char c='a'; c<='z'; c++){

char temp = arr[i];

if(arr[i]!=c){

arr[i]=c;

}

String newWord = new String(arr);

if(wordDict.contains(newWord)){

queue.add(new Word(newWord, top.numSteps+1));

wordDict.remove(newWord);

}

arr[i]=temp;

}

}

}

return 0;

}

}