**[One Edit Distance](https://leetcode.com/problems/one-edit-distance/)**

**public** **class** OneEditDistance {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println(*isOneEditDistance*("acb", "ab"));

}

**public** **static** **boolean** isOneEditDistance(String s, String t) {

**if**(s.equals(t)) {

**return** **false**;

}

**if**(Math.*abs*(s.length() - t.length()) >= 2) {

**return** **false**;

}

**for**(**int** i = 0 ; i < Math.*min*(s.length(), t.length()) ; i++) {

**if**(s.charAt(i) != t.charAt(i)) {

**if**(s.length() == t.length()) {

**return** s.substring(i + 1).equals(t.substring(i + 1));

}

**else** **if**(s.length() > t.length()) {

**return** s.substring(i + 1).equals(t.substring(i));

}

**else** {

**return** s.substring(i).equals(t.substring(i + 1));

}

}

}

**return** **true**;

}

}

Time complexity : O(n) , n is minimum of length of given words

Space Complexity : O(1) ,constant space