Our Solution(s) Run Code

Solution 1 Solution 2

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using namespace std;

// O(nm) time | O(min(n, m)) space
int levenshteinDistance(string str1, string str2) {
  string small = str1.length() < str2.length() ? str1 : str2;
  string big = str1.length() >= str2.length() ? str1 : str2;
  int *evenEdits = new int[small.length() + 1];
  int *oddEdits = new int[small.length() + 1];
  for (int j = 0; j < small.length() + 1; j++) {
    evenEdits[j] = j;
  }
  int *currentEdits;
  int *previousEdits;
  for (int i = 1; i < big.length() + 1; i++) {
    if (i % 2 == 1) {
      currentEdits = oddEdits;
      previousEdits = evenEdits;
    } else {
      currentEdits = evenEdits;
      previousEdits = oddEdits;
    }
    currentEdits[0] = i;
    for (int j = 1; j < small.length() + 1; j++) {
      if (big[i - 1] == small[j - 1]) {
        currentEdits[j] = previousEdits[j - 1];
      } else {
        currentEdits[j] = 1 + min(previousEdits[j - 1],
                                  min(previousEdits[j], currentEdits[j - 1]));
      }
    }
  }
  return big.length() % 2 == 0 ? evenEdits[small.length()]
                               : oddEdits[small.length()];
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
using namespace std;

int levenshteinDistance(string str1, string str2) {
  // Write your code here.
  return -1;
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.