

Брянцев Всеволод Александрович

ПММ, 4 курс, 61 группа

Отчет по лабораторной работе №3

Задание 1.

Входные данные: Две произвольные строки, содержащие буквы одного алфавита.

Выходные данные: Редакционное расстояние  $d(S1, S2)$ , используя алгоритм Вагнера-Фишера.

В качестве сравниваемых последовательностей взять последовательности S1 и S2 из Лабораторной работы 2.

Пример входных данных: S1 = PLEASANTLY, S2 = MEANLY

Пример выходных данных: 5

Код программы:

```
import 'dart:io';
import 'dart:async';

Future<void> main() async {
  String s1 = await File('bin\\input1.txt').readAsString();
  String s2 = await File('bin\\input2.txt').readAsString();

  s1 = s1.trim();
  s2 = s2.trim();

  final stopwatch = Stopwatch()..start();
  int result = wagnerFisherAlgorithm(s1, s2);
  stopwatch.stop();

  print('Редакционное расстояние: $result');
  print('Время выполнения: ${stopwatch.elapsedMilliseconds} мс');
}

int wagnerFisherAlgorithm(String s1, String s2) {
  if (s1.length > s2.length) {
    var temp = s1;
    s1 = s2;
    s2 = temp;
  }

  List<List<int>> matrix = List.generate(
    s2.length + 1,
    (i) => List.filled(s1.length + 1, 0),
  );

  for (int i = 0; i <= s1.length; i++) {
    matrix[0][i] = i;
```

```

}

for (int i = 0; i <= s2.length; i++) {
    matrix[i][0] = i;
}

for (int i = 1; i <= s2.length; i++) {
    for (int j = 1; j <= s1.length; j++) {
        if (s2[i - 1] == s1[j - 1]) {
            matrix[i][j] = matrix[i - 1][j - 1];
        } else {
            matrix[i][j] = 1 +
                [matrix[i - 1][j - 1], matrix[i - 1][j], matrix[i][j - 1]]
                    .reduce((a, b) => a < b ? a : b);
        }
    }
}

return matrix[s2.length][s1.length];
}

```

Файлы input1 и input2 содержат последовательности S1 и S2:

input1.txt

```

ttcgttaagtg ccctcaccat cctatgccat gccgcgcggc tgcctggtag ccctcggccc
tgcgcaatca tgaacacgca actaacaagc tcgcgttgta ggacaaggat ggcgatggtc
agtactcccc cttcgaactc acttccgcat actctgccct caaagcagcg ccgcatctcc
atccacgca atcggtaaag gggccggagc gaggcttgct ggctaggggt ccaaaacacc
gctcagagct acaagaccac gacatccgcc cgactcagta acaagcacia ctgacgatga
tgcgccacag gtcaaatcac caccaaggag ctaggcaccg tcatgcgctc gctcggccaa
aaccacagcg agtctgagct ccaggacatg atcaacgagg tcgatgccga caacaacggc
accattgact tcccaggtag atcctctcgt acgagtccaa cgtgccacag ctaacttctc
cagaattcct taccatgatg gcccgcaaga tgaaggacac cgactccgag gaggagatcc
gggaagcctt caaggtcttc gaccgcgata acaacggctt catctccgcc gccgaactgc
gtcacgtcat gacttctatt ggcgagaaat tgaccgatga cgaggtcgac gagatgatcc
gggaggctga ccaggacggt gacggccgta tcgactgtag gtcacaact

```

input2.txt

```

tcgagttcaa ggaggccttc tccctcttcg taagtagctc cctgtctgct tgacgcgcgg
ctgcctagag ccttgctgct aacaccatga ccatgttggt gctaacacat gttctctcta
taggacaagg atggcgatgg ttagtagtcc cccttcgaac tcacgcccgc gactctacc
cgaagccagc tcgcagtcct cggtctgcac gcgtctcaca ccagacaatc gctgaccatg
ttgcgccgca ggtcaaatca ccaccaagga gctcggaacc gtcatgcgct cgctcggcca
aaatcccagc gagtctgagc tacaggacat gatcaacgag gtcgacgccg acaacaacgg
caccattgac ttcccaggta cctcaattct ccgactacca actcaccaaa accctcacta
accacaaccc agagtctctc accatgatgg cccgcaagat gaaggacacc gactctgagg
aggagattcg ggaagccttc aaggtctttg accgtgacaa caacggcttc atctccgccg
ctgaactgcg tcacgtcatg acttctattg gcgaaaaatt gaccgatgac gaggttgacg
agatgatccg ggaggctgac caggacggcg acggccgcat cgactgtagg cgctaaagag
ttgctcggtt cttcaaaca ctaa

```

Результат для приведенных последовательностей:

```
Редакционное расстояние: 337
Время выполнения: 31 мс
```

## Задание 2.

Выполнить задание 1, уменьшив потребление памяти до  $O(\min\{m, n\})$ .

В качестве сравниваемых последовательностей взять последовательности S1 и S2 из Лабораторной работы 2.

Выписать время, которое потребовалось для выполнения программы в Задании 1 и в Задании 2 для сравнения последовательностей S1 и S2 из Лабораторной работы 2.

Код программы:

```
import 'dart:io';
import 'dart:async';

Future<void> main() async {
  String s1 = await File('bin\\input1.txt').readAsString();
  String s2 = await File('bin\\input2.txt').readAsString();

  s1 = s1.trim();
  s2 = s2.trim();

  final stopwatch = Stopwatch()..start();
  int result = wagnerFisherOptimized(s1, s2);
  stopwatch.stop();

  print('Редакционное расстояние: $result');
  print('Время выполнения: ${stopwatch.elapsedMilliseconds} миллисекунд');
}

int wagnerFisherOptimized(String s1, String s2) {
  if (s1.length > s2.length) {
    var temp = s1;
    s1 = s2;
    s2 = temp;
  }

  List<int> prevRow = List.generate(s1.length + 1, (i) => i);
  List<int> currRow = List.filled(s1.length + 1, 0);

  for (int i = 1; i <= s2.length; i++) {
    currRow[0] = i;
    for (int j = 1; j <= s1.length; j++) {
      if (s2[i - 1] == s1[j - 1]) {
```

```

        currRow[j] = prevRow[j - 1];
    } else {
        currRow[j] = 1 +
            [prevRow[j - 1], prevRow[j], currRow[j - 1]]
                .reduce((a, b) => a < b ? a : b);
    }
}
var temp = prevRow;
prevRow = currRow;
currRow = temp;
}

return prevRow[s1.length];
}

```

Здесь используются те же файлы input1 и input2 с последовательностями.

Результат для приведенных последовательностей:

```

Редакционное расстояние: 337
Время выполнения: 26 миллисекунд

```