

*Ведерников Николай Викторович,
Ульянцев Владимир Игоревич*

ЗАДАЧА «ПАЛИНДРОМЫ»

Этой статьей мы продолжаем цикл публикаций разборов олимпиадных задач для школьников по информатике. Решение таких задач и изучение разборов поможет вам повысить уровень практических навыков программирования и подготовиться к олимпиадам по информатике.

В статье рассматривается задача «Палиндромы», которая предлагалась на второй командной интернет-олимпиаде по программированию в 2012–2013 учебном году. Материалы этой олимпиады можно найти на сайте <http://neerc.ifmo.ru/school/io/>.

УСЛОВИЕ ЗАДАЧИ

После тяжелого футбольного матча Василий любит отдыхать, а именно смотреть



сериалы или играть в компьютерные игры. Но после очередного матча ему надоело играть в футбольные симуляторы и смотреть сериалы. Тогда он решил взять у своего брата-близнеца игру *Palin*.

Игра заключалась в следующем: дана строка S и k правил замены одного символа на другой. То есть, если i -й символ строки будет равен a_j , то его можно будет заменить на символ b_j , где a_j и b_j – символы j -го правила замены. Кроме того, если символ с номером i в строке уже был изменен, то изменить его еще раз невозможно. Целью игры является получение из данной строки палиндрома путем замен символов. Напомним, что палиндромом является строка, которая одинаково читается и слева направо и справа налево.

Василий задался следующим вопросом: за какое минимальное число операций замены такое возможно?

Формат входного файла

В первой строке входного файла дана строка S ($0 < |S| \leq 10^5$), содержащая только строчные буквы латинского алфавита. Во второй строке дано число k ($0 \leq k \leq 26$) – число правил. Следующие k строк содержат по два символа a_j и b_j (строчные буквы латинского алфавита), разделенных пробелом, – символы каждого из правил замены. Все символы a_j различны.

Формат выходного файла

Если из данной строки можно получить палиндром, следуя описанным правилам, то в первой строке выходного файла выведите минимальное число операций замены, необходимое для этого. Во второй строке в таком случае выведите номера символов, которые необходимо изменить. Символы в строке нумеруются с единицы.

Если же получить из данной строки палиндром невозможно, выведите в единственной строке выходного файла «-1».

Примеры входных и выходных данных

palindromes.in	palindromes.out
abc 1 a c	1 1
abc 1 b c	-1
abc 2 c d d a	-1

РАЗБОР ЗАДАЧИ

Так как из исходной строки необходимо получить палиндром, то мы должны сделать так, чтобы первый символ совпадал с последним, второй символ – с предпоследним и так далее. Таким образом, все символы разбиваются на пары (если строка была нечетной длины, то символ, стоящий в середине строки, находится в паре сам с собой).

Заметим, что для каждой пары задачу можно решать независимо от других пар. Покажем, как за минимальное число операций замены сделать так, чтобы каждая пара состояла из двух одинаковых символов.

Рассмотрим пару символов (x, y) . Если символ x равен символу y , то делать операцию замены не нужно, так как необходимо минимизировать число замен. Иначе, если символ x не равен символу y , нужно сделать замену одного или двух символов. Чтобы символы в паре стали одинаковыми за одну операцию замены, нужно, чтобы существовало такое правило с номером j , что либо символ a_j равен x , а символ b_j равен y , либо символ a_j равен y , а символ b_j равен x . Если же такого правила нет, то за одну операцию замены нельзя сделать символы равными.

Для того чтобы сделать символы равными за две замены, нужно, чтобы существовали правила с номерами j и l такие, что символ a_j равен x , символ a_l равен y и при этом символ b_j равен b_l . Если же таких правил нет, то сделать символы в этой паре одинаковыми невозможно, так как в паре не может быть более двух замен (к одному символу можно применить только одну операцию замены).

Если хотя бы одну пару символов нельзя сделать такой, чтобы они были одинаковы, то строку нельзя сделать палиндромом. В листинге 1 приведена программная реализация описанного алгоритма на языке программирования *Pascal*.

Листинг 1. Реализация алгоритма

```

const
  MAXN = 100000;
var
  i, k, cnt, len: integer;
  ans: array [1..MAXN] of integer;
  a: array ['a'..'z'] of char;
  x, space, y: char;
  s: string;
  possible: boolean;
begin
  assign(input, 'palindromes.in');
  reset(input);
  assign(output, 'palindromes.out');
  rewrite(output);

```

```
// Инициализируем правила замены пустыми правилами
for x := 'a' to 'z' do
  a[x] := '#';
readln(s);
readln(k);
// Считывание правил
for i := 1 to k do begin
  readln(x, space, y);
  a[x] := y;
end;

cnt := 0;
possible := true;
len := length(s);
for i := 1 to (len div 2) do begin
  x := s[i];
  y := s[len - i + 1];

  // Символы в паре совпали
  if (x = y) then
    continue;

  // Первый символ в паре заменяется на второй
  if (a[x] = y) then begin
    cnt := cnt + 1;
    ans[cnt] := i;
    continue;
  end;

  // Второй символ в паре заменяется на первый
  if (a[y] = x) then begin
    cnt := cnt + 1;
    ans[cnt] := len - i + 1;
    continue;
  end;

  // Символы в паре заменяются на одинаковые
  if ((a[x] = a[y]) and (a[x] <> '#')) then begin
    cnt := cnt + 1;
    ans[cnt] := i;
    cnt := cnt + 1;
    ans[cnt] := len - i + 1;
    continue;
  end;

  // Невозможно сделать символы в паре одинаковыми
  possible := false;
  break;
end;

// Вывод ответа
if (possible = false) then
  writeln('-1')
else begin
  writeln(cnt);
  if (cnt > 0) then begin
    for i := 1 to cnt - 1 do begin
```

```
        write(ans[i], ' ');  
    end;  
    writeln(ans[cnt]);  
end;  
end;  
close(input);  
close(output);  
end.
```

*Ведерников Николай Викторович,
студент третьего курса кафедры
«Компьютерные технологии» НИУ
ИТМО, член жюри Интернет-
олимпиад по информатике;*

*Ульянцев Владимир Игоревич,
студент шестого курса кафедры
«Компьютерные технологии» НИУ
ИТМО, член жюри Интернет-
олимпиад по информатике.*



Наши авторы, 2012.

Our authors, 2012.