**Федеральное Агентство по Образованию**

Государственное образовательное учреждение высшего профессионального образования

**Санкт-Петербургский Государственный Электротехнический Университет «ЛЭТИ» им. В.И.Ульянова (Ленина)**

**(СПБГЭТУ)**

Кафедра МОЭВМ

«Построение и анализ алгоритмов»

Лабораторная работа №3.

«Метод ветвей и границ»

Выполнил Богаевский Д.В.

Факультет КТИ

Группа № 1381

Преподаватель Казаков Б.Б.

#### Санкт-Петербург

# 2013

**Задание.**

**Задача 2.**

1. **«*Редакционное расстояние*».** Даны две строки *A* и *B*. Со строкой *A* разрешено выполнять следующие операции:

* удалить символ;
* вставить символ;
* заменить один символ на другой.

Какое наименьшее количество операций необходимо применить к строке *A*, чтобы получить строку *B*.

Примечание. Это количество операций принято называть *редакционным расстоянием.*

**Алгоритм решения данной задачи.**

1)Составляем алфавит символов, из которых состоит строка B.

2)Устанавливаем границу точного решения (добавление/удаление n символов + замена всех символов в слове).

3)Находим оптимальное решение.

А)Если длина строки А меньше длины B, то добавляем в каждое место каждый символ из алфавита и рекурсивно решаем новую задачу (увеличивая счетчик операций на 1).

Б)Если длина строки А больше длины В, то поочередно удаляем каждый символ и решаем задачи рекурсивно.

В)Если длина строки А равна длине В, то изменяем несовпадающие символы, затем сравниваем с текущим решением, если количество операций меньше, то заменяем текущее решение.

**Описание функций.**

char \*AnalyseB(char \*B); - составить алфавит из строки.

bool inMass(char \*MASS, char c); - содержится ли символ “c” в строке “MASS”.

void Solve(char \*A, char \*B, char \*alph, int n, int \*m); - решить данную задачу.

void add(char \*s, int n, int c); - добавить символ “c” в строку “s” на позицию “n”.

void del(char \*s, int n); - удалить символ из строки “s” с позиции “n”.

void change(char \*s,int n, int c); - заменить символ на “c” из строки “s” на позиции “n”.

char \*AnalyseB(char \*B); - составить алфавит из строки.

Исходные данные.

B – указатель на входную строку.

Рабочие переменные.

i – текущий символ во входной строке.

j – текущий символ в выходной строке.

Результаты.

res – алфавит.

Алгоритм.

1)Пока не конец строки B. Если символ не содержится в строке res, то добавляем его.

bool inMass(char \*MASS, char c); - содержится ли символ “c” в строке “MASS”.

Исходные данные.

MASS – указатель на строку.

с – символ.

Рабочие переменные.

i – номер текущего символа.

Результаты.

True – символ содержится в строке.

False – символ не содержится в строке.

void Solve(char \*A, char \*B, char \*alph, int n, int \*m); - решить данную задачу.

Исходные данные.

A – указатель на строку A.

B – указатель на строку B.

alph – указатель на алфавит.

n – количество совершенных операций.

m – указатель на найденное решение.

Рабочие переменные.

A2 – модифицированная одной операцией строка А.

i – номер места, в которое добавляем/удаляем символ.

j – номер символа в алфавите.

Результаты.

-

Алгоритм.

1)Если n>=m, то выходим из решения данной задачи.

2)Если длина строки А меньше длины строки В, то в каждое место добавляем каждый символ из алфавита и рекурсивно решаем новую задачу (увеличивая счетчик операций на 1).

3)Если длина строки А больше длины В, то поочередно удаляем каждый символ и решаем задачи рекурсивно.

4)Если длина строки А равна длине В, то изменяем несовпадающие символы, затем сравниваем с текущим решением, если количество операций меньше, то заменяем текущее решение.

void add(char \*s, int n, int c); - добавить символ “c” в строку “s” на позицию “n”.

Исходные данные.

s – указатель на строку.

n – позиция в строке.

c – символ.

Рабочие переменные.

i – номер текущего символа в строке.

Результаты.

-

Алгоритм.

1)Если позиция символа n превышает длину строки, то выходим из функции.

2)Сдвигаем все символы, начиная с n-го символа на 1 вправо.

3)Добавляем символ c на позицию n.

void del(char \*s, int n); - удалить символ из строки “s” с позиции “n”.

Исходные данные.

s – указатель на строку.

n – номер удаляемого символа.

Рабочие переменные.

i – номер текущего символа в строке.

Результаты.

-

Алгоритм.

1)Если позиция символа n превышает длину строки, то выходим из функции.

2)Сдвигаем все символы с позиции n+1 на 1 влево.

void change(char \*s,int n, int c); - заменить символ на “c” из строки “s” на позиции “n”.

Исходные данные.

s – указатель на строку.

n – номер символа в строке.

c – символ.

Рабочие переменные.

-

Результаты.

-

Алгоритм.

1)Если позиция символа n превышает длину строки, то выходим из функции.

2)Заменяем символ на позиции n на символ c.