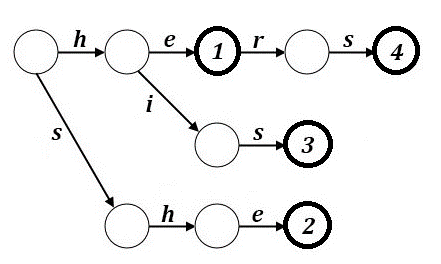
Лабораторная работа №8

**Бор**

**Бор** (англ. *trie*, луч, нагруженное дерево) – структура данных для хранения набора строк, представляющая из себя подвешенное дерево с символами на рёбрах. Строки получаются последовательной записью всех символов, хранящихся на рёбрах между корнем бора и терминальной вершиной. Размер бора линейно зависит от суммы длин всех строк, а поиск в бору занимает время, пропорциональное длине образца.

**Пример**: Бор для набора образцов {he, she, his, hers}:

[](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A4%D0%B0%D0%B9%D0%BB:%D0%91%D0%BE%D1%80.jpg)

Пусть *n* – это мощность алфавита, а *k* – это количество слов во множестве слов. Бор храним как набор вершин, у каждой из которых есть метка, обозначающая, является ли вершина терминальной, символ, который привел в данную вершину и указатели (рёбра) на другие вершины или на NULL.

struct vertex:

vertex next[n]

char ch

bool isTerminal

**Алгоритм построения бора для множества слов M={P0, P1,…, Pk}**

Начало.

Шаг 1. Создадим дерево из одной вершины (в нашем случае корня).

Шаг 2. Добавление элементов в дерево.

Добавляем слова *Pi* одно за другим. Следуем из корня по рёбрам, отмеченным буквами из *Pi*, пока возможно.

Если *Pi* заканчивается в *v*, отмечаем вершину *v* как терминальную.

Если ребра, отмеченного очередной буквой *Pi* нет, то создаем новое ребро и вершину для символа строки *Pi*.

Конец.

***Задание 1. Реализовать построение бора по заданному множеству слов М.***

***Задание 2. Поиск в боре.***

Дано множество слов М и слово S. Используя поиск в боре, определить, содержится ли слово S во множестве М.