Wstęp do programowania w języku C

Grupa MSz w poniedziałki

Lista 9 na zajęcia 9.12.2019

Zadanie 1. ***(15 punktów na pierwszej pracowni, 10 punktów na drugiej)***

Na wejściu dane są słowa złożone z liter 'a'–'z', każde w osobnym wierszu. Napisz program, który po wczytaniu danego słowa w wypisze liczbę takich słów u, które pojawiły się wcześniej i których prefiksem jest w. Do tego celu należy wykorzystać drzewo trie.

Można założyć, że długość słów jest ograniczona przez 10^6. Nie trzeba przejmować się potencjalnym brakiem pamięci na stercie ani liczbą wywołań rekurencyjnych.

Drzewo trie składa się z węzłów, z których każdy może mieć do 26 dzieci odpowiadających kolejnym literom. Każda ścieżka od korzenia do węzła reprezentuje słowo złożone z liter które wybieramy na krawędziach.
`https://pl.wikipedia.org/wiki/Drzewo_trie`

Przydatna definicja węzła:

```
typedef struct Node {
  int count;
  struct Node* edges[26];
} Node;
```

Węzeł drzewa zawiera więc wskaźniki do dzieci oraz licznik słów w drzewie, które mają prefiks odpowiadający ścieżce do tego węzła.

Przykład 1:

```
bb
aaa
bb
a
```

Wynik:

```
bb 0
aaa 0
```

```
bb 1
a 1
```

Przykład 2:

```
a
ab
abc
xyz
abc
ab
a
```

Wynik:

```
a 0
ab 0
abc 0
xyz 0
abc 1
ab 3
a 5
```

Wskazówka: Przykład jak można ręcznie utworzyć drzewo dla dwóch słów x, xy:

```
Node *root = (Node*)calloc(sizeof(Node),1);
root->edges['x'-'a'] = (Node*)calloc(sizeof(Node),1);
root->edges['x'-'a']->count = 2;
root->edges['x'-'a']->edges['y'-'a'] = (Node*)calloc(sizeof(Node),1);
root->edges['x'-'a']->edges['y'-'a']->count = 1;
```