

1 Домашнее задание №1

Срок исполнения - 23.09.2019

Задача №1 "Список"

Необходима создать динамическую структуру данных односвязный (кто хочет, может сделать двух-связанный) список. Он должен поддерживать следующие операции: - добавление нового узла в список - поиск по элементам списка - удаление элемента списка Поле полезной нагрузки должно хранить число. Так же необходимо создать функцию, которое будет принимать на вход число любой длины и представлять его в виде списка. Стандартные структуры типа list etc использовать нельзя.

Задача №2 "Длинная арифметика"

В качестве входящих данных у вас есть два связанных списка представляющих два не отрицательных числа. Цифры хранятся в обратном порядке и каждый элемент списка хранит одну цифру. Сложите два числа и верните результат в виде связанного списка. Предполагается что оба числа не содержат лидирующих нулей, кроме случая когда число само по себе 0. Числа могут быть сколь угодно большие.

Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)

Output: 7 -> 0 -> 8

2 Домашнее задание №2

Срок исполнения - 30.09.2019

Задача №1 "Строка — повторение подстроки"

Ввести непустую строку *s*. Найти такое наибольшее число *k* и такую строку *t*, что *s* совпадает со строкой *t*, выписанной *k* раз подряд. Вывести *k*.

Input: abcabcabcab

Output: 4

Задача №2 "Самое популярное слово"

Ввести построчно текст, состоящий из пробелов, переводов строки и латинских букв, и заканчивающийся пустой строкой. Вывести слово, которое чаще других встречается в тексте, если оно такое одно, и —, если таких слов несколько.

Input: Sed tempus ipsum quis eros tempus lacinia Cras finibus lorem ut lacinia egestas nunc nibh iaculis est convallis tincidunt mi mi sed nisl Sed porttitor aliquam elit ullamcorper tincidunt arcu euismod quis Mauris congue elit suscipit leo varius facilisis Cras et arcu sodales laoreet est vitae pharetra orci Integer eget nulla dictum aliquet justo semper molestie neque Maecenas bibendum lacus tincidunt auctor varius purus felis ullamcorper dui et laoreet ligula ex et risus Donec eget fringilla nibh Cras congue tincidunt accumsan Maecenas euismod eleifend elit ut rhoncus tortor sodales a Cras egestas finibus lorem non tempor tincidunt aera

Output: tincidunt

Задание №3 "Буквенные комбинации номера телефона"

На вход алгоритму подается строка, состоящая из цифр от 0 до 9 (номер телефона), нужно вернуть все возможные комбинации букв, которые могут быть представлены в номере.



Input: "23"

Output: ["ad "ae "af "bd "be "bf "cd "ce "cf"]

Задание №4 "Граф"

Реализовать структуру данных граф и два способа обхода графа:

- в глубину (DFS)
- в ширину (BFS)

На вход будет подан массив ребер графа $[[0, 3], [1, 3], [2, 3], [4, 3], [5, 4]]$ В результате должна быть структура, принимающая массив ребер и имеющая две функции, при вызове которых будет выполнять обход вершин. Посещаемые вершины должны выводиться на экран.

Задание №4 "Поиск пути"

Пусть дан граф, причем каждому его ребру сопоставлен вес (взвешенный граф). Требуется найти путь между двумя заданными вершинами с наименьшим весом.

На вход будет подан массив ребер графа $[[0, 3, 5], [1, 3, 11], [2, 3, 56], [4, 3, 77], [5, 4, 89]]$ Третий параметр - вес ребра.

У структуры данных граф должна быть функция, принимающая два параметра (nodeStart, nodeEnd). Данная функция должна вывести кратчайший путь из nodeStart в nodeEnd.

Задание №5 "Время задержки сети"

У нас есть сеть состоящая из узлов, каждый из которых помечен меткой от 1 до N . Дан список времени распространения сигнала от одного узла до другого в виде направленных ребер. $times[i] = (u, v, w)$, где u - начальный узел, v - конечный узел, и w - время распространения сигнала от начального узла к конечному. Мы отправляем сигнал из узла X и хотим узнать как долго он будет распространяться сигнал до всех узлов. Если это не возможно, то возвращаем -1.

Input: times = $[[2,1,1],[2,3,1],[3,4,1]]$, $N = 4$, $X = 2$

Output: 2

Дополнительные условия:

1. $N \in [1, 100]$
2. $X \in [1, N]$
3. $times \in [1, 6000]$
4. Все ребра $times[i] = (u, v, w)$ будут иметь $1 \leq u, v \leq N$ и $0 \leq w \leq 100$