

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Информационные Системы

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
Тема: Списочные структуры

Студентка гр. 1373

Зашляхтин А. А.

Преподаватель

Бондаренко Б.Е.

Санкт-Петербург

2022

Цель работы.

Реализовать следующие структуры: односвязный список, динамический массив и стек. Стек можно реализовать как на базе списка, так и отдельно. Использовать стек для реализации алгоритма сортировочной станции. Разрешённые символы в исходном выражении: +, -, *, /, ^, sin, cos, (,), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Для упрощения разбиения входной строки на токены разрешается отделять каждый символ пробелом.

Ход работы

В процессе работы были реализованы следующие структуры данных:

- Динамический массив
- Односвязный список
- Стек

➤ *Динамический массив*

Основное отличие динамического массива от обычного статического массива – это возможность изменяться во время исполнения программы.

В моей программе динамический массив имеет следующие методы:

1. Доступ к элементу массива по индексу (по сложности занимает $O(1)$)
2. Поиск элемента в массиве по значению (по сложности занимает $O(n)$)
3. Добавление элемента по индексу (по сложности занимает $O(n)$)
4. Удаление элемента из массива по индексу (по сложности занимает $O(n)$)

➤ *Односвязный список*

Представляет собой структуру данных в виде отдельного элемента с информацией, скрепленного с последующим таким же элементом в одностороннем порядке указателям.

В моей программе односвязный список имеет следующие методы:

1. Доступ к элементу списка по индексу (по сложности занимает $O(n)$)
2. Поиск элемента в списке по значению (по сложности занимает $O(n)$)
3. Добавление элемента по индексу (по сложности занимает $O(n)$)
4. Удаление элемента из списка по индексу (по сложности занимает $O(n)$)

➤ **Стек**

Это такой тип данных, реализованный по принципу «последний вошел, первый вышел» или LIFO. (например: стопка книг; чтоб взять книгу из середины или конца, для начала нужно достать книги выше нужной)

В моей программе односвязный список имеет следующие методы:

1. Добавление элемента на вершину стека (по сложности занимает $O(1)$)
2. Снятие вершины стека (по сложности занимает $O(1)$)
3. Возврат верхнего элемента стека без его снятия (по сложности занимает $O(1)$)
4. Удаление элемента из стека (по сложности занимает $O(1)$)

С помощью результата работы над структурами данных был реализован алгоритм сортировочной станции.

Принцип работы:

- Принимает на вход математическое выражение в инфиксной форме
- Выдает на выходе выражение в постфиксной форме (обратной польской записи)

Примеры работы:

1. Правильное выражение

Microsoft Visual Studio Debug Console

```
Enter expression in classic form(use numbers and +, -, *, /, ^, (, ), sin, cos,):  
( 5 + 3 ) * 8 + sin ( 2 ^ 5 ) / 6 - ( 8 + 3 )  
Result is:  
5 3 + 8 * 2 5 ^ sin 6 / + 8 3 + -
```

2. Неправильное выражение

C:\Users\Артём\Desktop\АИСД\LAB1\Debug\LAB1.exe

```
Enter expression in classic form(use numbers and +, -, *, /, ^, (, ), sin, cos,):  
( 2 + 3 ) * 3 - ( 6 + 3  
Result is:  
Wrong number of stamples!
```

Вывод

В результате работы были реализованы следующие структуры данных:

- 1) Односвязный список;
- 2) Динамический массив;
- 3) Стек.

На основе реализованных структур, был написан алгоритм сортировочной станции, переводящий выражение из инфиксной записи в постфиксную запись.