

# Лабораторная работа А: Связные списки

## Лабораторная работа А: Связные списки

### Списки

[Вариант 10: Стек \(1\)](#)

[Вариант 11: Очередь \(1\)](#)

[Вариант 12: XOR связный список \(2\)](#)

[Вариант 13: Развёрнутый связный список \(3\)](#)

[Вариант 14: Списки с пропусками \(3\)](#)

[Вариант 15: JumpList \(4\)](#)

[Вариант 16: Полиномы \(3\)](#)

[Вариант 17: Длинная арифметика \(3\)](#)

[Вариант 18: Сортировка односвязного списка \(2\)](#)

[Вариант 19: Сортировка двухсвязного списка \(2\)](#)

[Вариант 110: Конкатенация строк \(1\)](#)

### Упорядоченные списки

[Вариант 20: Очередь с приоритетами \(1\)](#)

[Вариант 21 \(1\)](#)

[Вариант 22: Список дней рождений \(1\)](#)

[Вариант 23: Длина слов \(1\)](#)

[Вариант 24: Трудовое расписание \(2\)](#)

[Вариант 25: Записная книжка \(2\)](#)

[Вариант 26: Температура \(2\)](#)

[Вариант 29: Дерево папок \(3\)](#)

### Операции с множествами

[Вариант 30: Множества \(2\)](#)

[Вариант 31: Слияние множеств \(2\)](#)

[Вариант 32: Соединение слиянием упорядоченных списков \(2\)](#)

### Разное:

[Вариант 40: Самоорганизующиеся списки \(3\)](#)

## Списки

---

### Вариант 10: Стек (1)

---

- Реализуйте [стек?](#) с помощью связного списка
- Реализуйте [стек?](#) с помощью массива
- Сравните полученные реализации, в чем преимущества и недостатки каждой?

### Вариант 11: Очередь (1)

---

- Реализуйте [очередь?](#) с помощью связного списка
- Реализуйте [очередь?](#) с помощью массива
- Сравните полученные реализации, в чем преимущества и недостатки каждой?

### Вариант 12: XOR связный список (2)

---

[XOR-связный список](#) — структура данных, похожая на обычный двусвязный список, однако в каждом элементе хранящая только один адрес — результат выполнения операции XOR над адресами предыдущего и следующего элементов списка. Для того, чтобы перемещаться по списку, необходимо взять

два последовательных адреса и выполнить над ними операцию XOR, которая и даст реальный адрес следующего элемента.

- Реализуйте XOR-связный список элементами которого являются ASCII-Z строки, со следующими операциями:
  - Добавить элемент в список
  - Удалить элемент из списка по ключу
  - Удалить данный элемент списка (по адресу элемента)
  - Найти элемент по ключу
  - Итерирование по списку
- В чем недостатки и преимущества XOR-связанного списка?

### Вариант 13: Развёрнутый связный список (3)

Развёрнутый связный список — список, каждый физический элемент которого содержит несколько логических (обычно в виде массива, что позволяет ускорить доступ к отдельным элементам).

- Реализуйте развернутый связный список, элементами которого являются целые числа со следующими операциями
  - Добавить элемент
  - Найти номер элемента по ключу
  - Найти элемент с заданным номером
  - Удалить элемент с заданным номером
  - Подсчитать количество элементов
  - По заданному элементу найти следующий (итерирование)
- Обратите особое внимание на интерфейсы вашей системы
- В чем преимущества и недостатки таких списков?

### Вариант 14: Списки с пропусками (3)

Реализуйте списки с пропусками с тремя основными операциями: Вставка, Удаление и Поиск элемента по ключу. Сравните производительность списков с пропусками с обычным линейным списком. Для этого замерьте время выполнения (большого числа) операций вставки, удаления и поиска элементов.

### Вариант 15: JumpList? (4)

Реализуйте JumpList? с операциями построение Jump-списка по заданному списку, поиска элемента, добавления и удаления элемента из JumpList? . Сравните производительность таких списков с обычным односвязным списком. Для этого замерьте время выполнения (большого числа) операций вставки, удаления и поиска элементов.

### Вариант 16: Полиномы (3)

Напишите программу, складывающую умножающую и дифференцирующую полиномы.

В текстовом файле 'input.txt' записаны два полинома, по одному в каждой строке. Каждый моном полинома имеет вид 'M\*x^k' или 'M', где: M – это некоторое целое число по модулю меньше 215, k – натуральное число, не превосходящее 230. Например, '10\*x^2'. В полиноме мономы не обязательно упорядочены по степеням. Выведите в текстовый файл 'output.txt' три полинома, по одному в каждой строке. Первый полином должен представлять сумму двух исходных, второй – их произведение, третий – производную первого полинома. В выходном файле полином должен быть упорядочен по степеням, начиная с младших ( $x^0$ ). Полином должен быть записан корректно с точки зрения принятых обозначений в математике:  $x^0$  – это 1, вместо  $x^1$  написан x, не должно быть выражений вида 'x+-3', вместо '1\*x' должно быть просто 'x'

Пример 'input.txt'	Пример 'output.txt'
$-3+x^2-7*x$	$-2-6*x+x^2$
$x+1$	$-3-10*x-6*x^2+x^3$
	$-7+2*x$

## Вариант 17: Длинная арифметика (3)

Напишите функции сложения, умножения и возведения в степень для длинных целых чисел.

## Вариант 18: Сортировка односвязного списка (2)

Пусть вам дан односвязный список, элементами которого являются ASCII-Z строки. Вам требуется отсортировать данный список.

- Придумайте и реализуйте алгоритм сортировки в данном случае. Какова его алгоритмическая сложность?

## Вариант 19: Сортировка двухсвязного списка (2)

Пусть вам дан двухсвязный список, элементами которого являются ASCII-Z строки. Вам требуется отсортировать данный список.

- Придумайте и реализуйте алгоритм сортировки в данном случае. Какова его алгоритмическая сложность?

## Вариант 110: Конкатенация строк (1)

Вам дан односвязный линейный список, элементами которого являются строки произвольной длины. Объедините эти строки в одну. Предложите наиболее эффективную реализацию.

## Упорядоченные списки

## Вариант 20: Очередь с приоритетами (1)

Реализуйте очередь с приоритетами с помощью отсортированного связного списка. Какова сложность операций в данном случае?

## Вариант 21 (1)

В текстовом файле заданы следующие данные (построчно): слово и некоторое ключевое число, соответствующее этому слову:

```
Слово1      20
Слово2      86
...
```

Считать файл в список таким образом, чтобы он оставался отсортированным по ключевым числам. Вывести список слов и соответствующих им ключевых чисел на экран. Если два слова в списке имеют одно и тоже значение ключевого числа, выдать сообщение об ошибке. Ввести с клавиатуры ключевое число и проверить, есть ли в списке слово, соответствующее такому ключевому числу.

## Вариант 22: Список дней рождений (1)

Файл содержит следующую информацию (построчно): фамилия, имя, отчество, дата рождения. Считать файл в список таким образом, чтобы он оставался отсортирован по дате рождения. Вывести список фамилий и дат рождения на экран (отсортированным по датам рождения). Ввести с клавиатуры дату рождения и проверить, есть ли в списке человек (один или несколько), родившийся в этот день.

### Вариант 23: Длина слов (1)

---

Считать все слова из текстового файла в список таким образом, чтобы список слов оставался отсортированным по длине слов (при совпадении длины следует сортировать список по алфавиту). Вывести на экран список всех встречающихся в файле слов, длина которых превышает N символов. Ввести с клавиатуры число и проверить, есть ли в списке слова такой длины. Вывести эти слова на экран в алфавитном порядке

### Вариант 24: Трудовое расписание (2)

---

Файл содержит следующую информацию (построчно): дата, фамилия и имя, количество рабочих часов. Считать информацию из файла в список таким образом, чтобы он оставался отсортирован по количеству рабочих часов (при совпадении количества рабочих часов сортировать список по алфавиту). Вывести на экран информацию о тех сотрудниках (фамилия и дата), которые суммарно отработали более N часов

### Вариант 25: Записная книжка (2)

---

Файл содержит следующую информацию (построчно): фамилия, имя, отчество. Считать файл в список таким образом, чтобы он оставался отсортирован по фамилиям (при совпадении фамилий - по именам; при совпадении и фамилий и имен - по отчествам). Вывести список фамилий на экран. Ввести с клавиатуры начальные буквы фамилии, имени и отчества и проверить, есть ли люди с такой фамилией в списке. Пример:

Иванов Иван Иванович

Петров Иван Петрович

- Запрос «И» «И» «» выдает первую фамилию в списке
- Запрос «» «Ив» «» выдает обе фамилии в списке

### Вариант 26: Температура (2)

---

Файл содержит следующую информацию (построчно): дата и средняя температура в этот день. Считать файл в список таким образом, чтобы он оставался отсортированным по возрастанию средней температуры (при совпадении средней температуры проводить сортировку следует по дате). Вывести на экран информацию обо всех днях, в которых средняя температура оказалась ниже нуля. Ввести с клавиатуры число и проверить, есть ли в списке день с такой средней температурой

### Вариант 29: Дерево папок (3)

---

По заданному списку путей восстановите дерево папок.

В первой строке текстового файла input.txt написано число натуральное число N ( $1 \leq N \leq 500$ ), в последующих N строках записаны пути к папкам. Каждый путь занимает отдельную строку, пробелы, включая пробелы в начале и конце строки, отсутствуют. Длина пути не превышает 255 символов. Имена папок разделены символом «\».

Имя каждой папки не превышает 8 символов. Оно может состоять из букв латинского алфавита "a-zA-Z", цифр "0-9", а также набора специальных символов "!#\$%&'()-@^\_`{}~".

Выведите в текстовый файл output.txt дерево каталогов, в следующем формате: Имя каждого каталога должно быть выведено на отдельной строке, перед именем должно стоять пробелы количеством, соответствующим уровню вложенности папки. Подкаталоги должны быть выведены сразу же за их родителем в лексикографическом порядке, с отступом на один пробел большим, чем у родителя. Поддиректории корневого каталога должны быть выведены в лексикографическом порядке без отступа.

Пример input.txt:

```
7
WINNT\SYSTEM32\CONFIG
GAMES
WINNT\DRIVERS
HOME
WIN\SOFT
GAMES\DRIVERS
WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86
```

Пример: output.txt

```
GAMES
DRIVERS
HOME
WIN
  SOFT
WINNT
  DRIVERS
    SYSTEM32
      CERTSRV
        CERTCO~1
          X86
CONFIG
```

## Операции с множествами

---

### Вариант 30: Множества (2)

---

Одним из возможных представлений множеств? является представление с помощью упорядоченных списков. Реализуйте с помощью связанных списков следующие алгоритмы работы с множествами:

- Добавить/Удалить элемент из множества
- Проверить есть ли элемент в множестве
- Объединить два множества
- Пересечь два множества

### Вариант 31: Слияние множеств (2)

---

В двух текстовых файлах записаны слова. Считайте слова из каждого файла в отдельный список таким образом, чтобы каждый список оставался отсортированным по алфавиту. Объедините два отсортированных списка в один упорядоченный список. Исходные данные: В файле input1.txt и в файле input2.txt содержатся слова из английских букв по одному в каждой строчке. Длина слова не превышает 20 символов. Считайте эти слова в два упорядоченных по алфавиту списка, объедините эти два списка в один упорядоченный список. Выведите полученный результат в файл output.txt Пример:

input1.txt	input2.txt	output.txt
word	banana	abc
abc	force	banana
realloc	create	create
	destroy	destroy
		force
		realloc

## Вариант 32: Соединение слиянием упорядоченных списков (2)

При работе с базами данных часто приходится выполнять операцию соединения для двух и более таблиц. Реализуйте алгоритм соединения для 2х таблиц слиянием упорядоченных списков. Исходные данные храните во внешней памяти.

Разное:

## Вариант 40: Самоорганизующиеся списки (3)

На примере операции поиска реализуйте самоорганизующиеся связанные списки. Используйте две стратегии:

- Перестановка найденного элемента в начало списка
- Перестановка найденного элемента на один уровень вверх

Организируйте длинную последовательность неслучайных запросов поиска в список (одни элементы ищутся чаще других). Оцените время работы операции поиска. Сравните время поиска с временем поиска в обычном связанном списке.

- Set ALLOWTOPICCHANGE = [TWikiLecturersGroup](#)

-- [OlegTalalov](#) - 16 Aug 2010

Topic revision: r13 - 02 Sep 2013 - 08:48:34 - [OlegTalalov](#)

Copyright © by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? [Send feedback](#)

