Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра информационных технологий и систем

**ОБРАБОТКА СПИСКОВ И РЕКУРСИЯ В ЯЗЫКЕ PROLOG**

Лабораторная работа по дисциплине:

“Функциональное и логическое программирование”

Выполнил студент группы 8091:

\_\_\_\_\_\_\_\_\_\_ / Васильев И.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Принял преподаватель:

\_\_\_\_\_\_ / Михайлов Д.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**Великий Новгород**

**2021**

# Цель и задачи

Целью работы является изучение приемов работы со списками в Прологе, а также более детальное изучение рекурсивного программирования применительно к обработке списков.

Задача №1  
Написать программу сортировки списка методом Шелла.

Вычисление шага выполняется на основе метода Р. Седжвика

Задача №2  
Написать программу сортировки методом прямого выбора.  
  
Задача №3  
Написать программу решения головоломки «Пирамида из домино».

Задача №4

Написать программу включения в исходный список элементов другого списка с заданным интервалом i.

# Решение

Для решения задач была выбрана среда Visual Prolog 5.2

Задача №1  
Сортировка Шелла это, по-сути, модификация схем сортировки других алгоритмов. Фактически для сортировки элементов используются другие алгоритмы, такие как: пузырьком, вставками, выбором и т.д. Но только эти алгоритмы применяются не ко всей исходной последовательности, а к ее частям.

Сначала в исходной последовательности сортируются между собой элементы, отстоящие друг от друга на расстоянии n/2 элементов, затем на расстоянии n/4 и т.д. до тех пор пока не получим 2 последовательности, элементы которых отстоят друг от друга на расстоянии 1-го элемента. После этого делаем сортировку этой полученной последовательности выбранным методом и на выходе имеем уже полностью отсортированную последовательность.

Возникает вопрос: зачем же были предыдущие сортировки? Для того, чтобы расположить сортируемые элементы наиболее близко к своим положенным позициям. А в этом случае в последней сортировке по всей последовательности значительно сокращается количество перестановок.

Приращение в сортировке Шелла - это расстояние между сортируемыми элементами динамически меняющееся на каждом проходе. Главное требование, чтобы на последней итерации оно было равно 1. Динамика изменения этой величины очень существенно сказывается на производительности алгоритма в целом.

Очевидно, что программист может выбрать любой алгоритм уменьшения этого приращения на каждом шаге, главное, чтобы в конце оно приняло значение 1. Существует немало стратегий рассчета приращения на каждом проходе алгоритма Шелла.

Например, Р. Седжвик, предложил такую схему вычисления прирашений:  
d[i] = 9\*2i - 9\*2i/2 + 1, если i четно  
d[i] = 8\*2i - 6\*2(i+1)/2 + 1, если i нечетно

Было доказано, что используя эту схему производительность алгоритма возрастает ~ O(n7/6) в среднем и до ~ O(n4/3) в худшем случае. При расчете приращений по этому методу останавливаться следует на значении d[i-1], если 3\*d[i] > n. Обычно расчет начинается с нулевых значений i(=[0,1,2...]) и продолжается до такого i, когда 3\*d[i+1] > n, как было сказано ранее. Т.о. данная процедура рассчета запускается перед самой сортировкой Шелла и затем хранит полученную таблицу приращений в памяти, а алгоритм сортировки на каждом шаге к ней обращается за очередным значением.

Программный код представлен в файле, приложенный к архиву лабораторной работы под именем “Lab61.pro”

Результат выполнения программы:

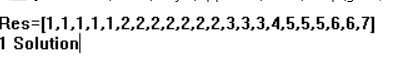


Рисунок Результат выполнения программы при: 2, 1, 2, 5, 3, 2, 1, 6, 5, 4, 3, 2, 1, 7, 6, 2, 1, 2, 5, 3, 2, 1

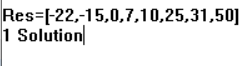


Рисунок Результат выполнения программы при: 10, 50, 25, 31, -15, -22, 7, 0

При решении задачи использованы списки и принципы работы с ними в Прологе. Также использованы математические функции, поддерживаемые Прологом.

В разделе goal сформулирована цель поиска – это список Res.

# Задание 2

Шаги алгоритма:

1. находим номер минимального значения в текущем списке
2. производим обмен этого значения со значением первой неотсортированной позиции (обмен не нужен, если минимальный элемент уже находится на данной позиции)
3. теперь сортируем хвост списка, исключив из рассмотрения уже отсортированные элементы

Программный код представлен в файле, приложенный к архиву лабораторной работы под именем “Lab62.pro”

Результат выполнения программы:

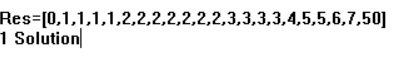


Рисунок Результат выполнения программы при: 2, 1, 2, 5, 3, 2, 1, 0, 5, 4, 3, 2, 3, 7, 6, 2, 1, 2, 50, 3, 2, 1



Рисунок Результат выполнения программы при: 10, 50, 25, 31, -15, -22, 7, 0, 10, 15

# Задание 3

Написать программу решения головоломки «Пирамида из домино».

Требуется расположить комплект домино в виде пирамиды, соблюдая следующие условия:

– В каждой строчке сумма очков на косточках должна быть точным квадратом.

– В строчках косточки укладываются согласно правилам игры в домино: 0 к 0, 1 к 1 и т. д.

Программный код представлен в файле, приложенный к архиву лабораторной работы под именем “Lab63.pro”. Там же представлены комментарии к коду.

Результат выполнения программы:

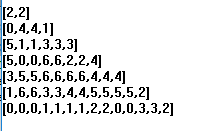


Рисунок Результат выполнения программы, исходный список: полный набор костяшек

При разработке программы используются списки и метод рекурсии для работы с ними.

В разделе goal описывается цель поиска. Domino – конкретизированный список, где каждая пара чисел соответствует кости домино. Переменные Str1, Str2, Str3, Str4, Str5, Str6, Str7 – это переменные, значения которых должны быть найдены в ходе выполнения программы.

При выводе результата каждая пара чисел также соответствует кости.

# Задание 4

Написать программу включения в исходный список элементов другого списка с заданным интервалом i.

Программный код представлен в файле, приложенный к архиву лабораторной работы под именем “Lab64.pro”.

Результат выполнения программы:



Рисунок Результат выполнения программы: исходный список: [1,2,3,4], второй список:[5,6,7], интервал: 2

Обоснование:

В решении задачи использованы списки и принципы работы с ними в языке Пролог.

В разделе goal целевая переменная, которая должна быть конкретизирована в ходе выполнения программы – List. Третий аргумент – интервал вставки элементов.

# Выводы по проведенным экспериментам

В ходе выполнения задач лабораторной работы, я изучил приемы работы со списками в языке Пролог и применил принципы рекурсии для обработки списков.