

## **Тема для команды 1: Хеширование данных во внешней памяти**

(имеются ввиду методы хеширования, которые можно использовать для больших объемов данных в файле. Это такие методы, как виртуальное, линейное, спиральное хеширования).

## **Тема для команды 2: Организация хранения и обработки данных о перемещениях объекта в пространстве**

(например, индексирование траектории перемещения автомобиля, или курьера. Есть специальные деревья, предназначенные для этого. В книге Василия Константиновича и т. д. Такие структуры описаны).

## **Тема для команды 3: Методы прогнозирования положения объекта в пространстве**

(аналогично предыдущей теме речь идет о перемещении, например, автомобиля по дорогам. Только упор в этой теме нужно сделать на расчете траектории перемещения (поиск пути в графе) и предсказании положения объекта, зная его скорость и вектор движения).

### **Роли в командных темах:**

- **РМ** — отвечает за весь проект, сроки, календарный план.
- **Аналитик** — анализ предметной области, выбор методов и подходов, анализ применений.
- **Проектировщик** — общий проект, структура модулей, классов, методов. Блок-схемы алгоритмов, псевдокод. В нашем случае на него возлагается и документация.
- **Кодер** — разработка библиотеки по проекту. Отмечает все просчеты проектировщика. Создает законченный продукт (в нашем случае возможно без интерфейса). Проверка работоспособности и первичное функциональное тестирование так же возложено на него.
- **Тестировщик** — разработка планов нагрузочных тестов и тестов в крайних ситуациях. Так же разрабатывает утилиты для генерации или получения из открытых источников данных для тестов. Проводит тестирование под нагрузкой в разных условиях, стоит графики.
- **Исследователь** — основная задача — сравнить методы между собой, исследовать их слабые и сильные стороны, показать как влияют разные параметры друг в самих алгоритмах или входные данные. Использует утилиты тестировщика для генерации тестовых данных (возможен случай, когда он сам разрабатывает утилиты генерации данных, а тестировщик использует эти утилиты (по усмотрению РМ)).
- **Внедренец** — дописывает визуальную часть для использования библиотеки в конкретном применении. Описывает возможные применения библиотеки и обосновывает эффективность её применения.

## **Исследовательские темы на 5+:**

### **1. Сравнение эффективности методов хеширования с динамической и статической функциями**

#### **Реализовать:**

- классическая схема хеширования делением с разрешением коллизий методом цепочек;
- виртуальное хеширование чисел в памяти (вариант В.Литвина).

#### **Провести следующие испытания:**

- как ведут себя обе схемы при увеличении количества записей («процент использования памяти», «число обращений к блокам»).

#### **Дополнительно изучить и описать:**

- анализ работ основателя виртуального хеширования Витольда Литвина (проанализировать все работы, для каждой работы составить аннотацию (краткое описание про что статья, можно на основе свободнодоступных аннотаций), что возможно - попробовать найти и скачать)
- примеры эффективного использования

#### **Обязательная литература:**

- [+] Virtual Hashing. A Dynamically Changing Hashing.

### **2. Сравнение эффективности методов хеширования с динамической и статической функциями**

#### **Реализовать:**

- генератор случайных чисел с разными характеристиками распределения (равномерное, нормальное, кластерное)
- виртуальное хеширование чисел в памяти (вариант В.Литвина)
- усовершенствованную разновидность виртуального хеширования применительно для чисел. Хеширование происходит как обычно. Однако после продолжительного расширения (если это окажется возможно) заменить начальный размер хеш-таблицы  $N$  на  $2N$ . Это увеличит размер флагов, но приведёт к уменьшению нагрузки при дальнейшем расширении.

#### **Провести следующие опыты:**

- число дисковых операций доступа и процент использования памяти при разных коэффициентах переполнения (4-6 графиков; по оси ОХ – количество записей в индексе; по оси ОУ – количество операций доступа / коэффициент заполнения)
- число дисковых операций доступа и процент использования памяти при разных распределениях данных (равномерное распределение, нормальное, кластерное)
- аналогичные опыты для улучшенного метода (на сколько улучшение сказалось на показателе «коэффициент использования памяти», «число доступов к диску», «число операций при перехешировании (расширении)»).

#### **Дополнительно изучить и описать:**

- функции  $VH0, VH1, VH2$

#### **Обязательная литература:**

- [+] Virtual Hashing. A Dynamically Changing Hashing.

### **3. Хеширование с частичным расширением**

#### **Реализовать:**

- Линейное хеширование (LH).
- Линейное хеширование с частичным расширением (LHPE).

#### **Провести следующие испытания:**

- Как сказывается принцип частичного расширения на эффективности (построить график зависимости количества операций от объема данных)
- Как сказывается принцип частичного расширения на использовании памяти.

#### **Дополнительно изучить и описать:**

- Проанализировать другие варианты линейного хеширования (полное изучение не требуется, достаточно привести обзор с краткой характеристикой (просто взятой из аннотации))

#### **Обязательная литература:**

- [+] Liitwin, W.: 'Linear hashing: a new tool for file and table addressing',
- [+] Larson, P.-A.: 'Linear hashing with partial expansions',

#### 4. Спиральное хеширование

**Реализовать:**

- Спиральное хеширование.
- Линейное хеширование (возможно взять результаты у товарища, реализующего предыдущий вариант)

**Провести следующие испытания:**

- как внедрение принципа неравномерного заполнения сказывается на таких показателях, как «количество операций» и «коэффициент использования памяти»

**Дополнительно изучить и описать:**

- Проанализировать другие варианты хеширования с неравномерной схемой заполнения (полное изучение не требуется, достаточно привести обзор методов с краткой характеристикой (возможно просто взятой из аннотации статей))

#### Темы на 5:

- Поиск кратчайшего пути во взвешенном графе
- Поиск минимального оставного дерева в графе
- Поиск максимального потока в графе
- Линейное хеширование в файле
- Расширяемое хеширование в файле
- Виртуальное хеширование в файле
- Спиральное хеширование в файле
- HR+-дерево
- TPR\*-дерево
- PMR-quadtrees для движущихся объектов
- MV3R-дерево
- TB-дерево
- STAR-дерево
- PR-дерево
- 2-3 TR-дерево
- 2+3 R-дерево
- VA-файл
- PCFI+индекс
- SV-model
- PSI
- MTSB-дерево
- GS-дерево
- BBx-индекс
- Bx-дерево
- STP-дерево
- RT-дерево
- MR-дерево
- LUGrid
- 3D R-дерево
- ST2B-дерево
- Bdual-дерево
- MOVIES
- IMORS
- RUM-дерево

- LUR-дерево
- POLAR-дерево
- PO-дерево
- R+-дерево
- R\*-дерево
- SR- деревья
- X-дерево
- TV-дерево
- OS-дерево

#### **Темы с большим упором на поиск и изучение материала (на 5):**

(это не избитые темы, материала здесь не много и в большинстве случаев на eng. Однако в таких темах, при условии что студент с темой *разобрался*, он получает +1бал. Т.е. даже без написанного приложения можно получить 4).

- LH\* : Linear Hashing for Distributed Files
- LH\*lh: A Scalable High Performance Data Structure for Switched Multicomputers
- LH\*S : a high-availability and high-security Scalable Distributed Data Structure
- Локально-чувствительное хеширование LSH (Locality Sensitive Hashing)
- Семантическое хеширование
- Спектральное хеширование
- Геометрическое хеширование
- Каскадное хеширование

#### **Простые темы:**

- Поиск кратчайшего пути в графе
- Алгоритмы обхода графа
- Интерполяционный поиск
- Бинарный поиск
- Внешняя сортировка
- Сортировка вставками
- Обменные сортировки
- Сортировки выбором
- Классические схемы хеширования
- Кодовые деревья, оптимальные префиксные коды
- Деревья И/ИЛИ
- Нагруженных деревья
- Красно-черные деревья
- АВЛ-деревья
- Прошитое бинарное дерево
- PATRICIA деревья
- (2-3) – деревья
- В-деревья
- Деревья промежутков
- Суффиксные деревья
- R-дерево
- K-D-дерево
- Qadro-дерево