

### Контрольные вопросы:

- ☐ (5 б.) Когда используются контейнеры типа (мульти) множество и отображение?
- ☐ (5 б.) Каким требованиям должна удовлетворять качественная хэш-функция?
- ☐ (5 б.) Из-за чего в хэш-таблицах возникают коллизии и как можно их разрешать?
- ☐ (5 б.) Почему сложность основных операций хэш-таблиц в худшем случае  $O(N)$ ?
- ☐ (5 б.) Что позволяет сделать инструмент создания контейнеров Boost.Multiindex?

### Упражнения:

- ☐ (25 б.) Операция вставки в контейнер `std::set` имеет сложность  $O(\log N)$ . Сгенерируйте  $N$  случайных чисел и добавьте их в данный контейнер. Совокупность этих операций имеет сложность  $O(N \log N)$ . Далее создайте последовательный контейнер, добавьте в него те же  $N$  случайных чисел и отсортируйте его с помощью алгоритма `std::sort`. Совокупность этих операций также имеет сложность  $O(N \log N)$ . Различие будет только в константе. Используя таймер, определите, что выполняется быстрее. Рассмотрите в качестве последовательного контейнера `std::vector` как вариант по умолчанию и `std::array` как самый быстросортируемый контейнер. Также определите, существует ли такое пороговое значение  $N$ , при котором сменяется лидер.
- ☐ (25 б.) Предложите хэш-функцию для хэширования чисел с плавающей точкой и определите ее свойства.
- ☐ (25 б.) На семинаре мы рассмотрели хэш-функцию из Boost, применяемую для вычисления хэш-кода по набору значений. Исследуйте равномерность данной хэш-функции. Создайте структуру с полями различных типов, сгенерируйте достаточно большое количество случайных уникальных экземпляров данной структуры и для каждого вычислите хэш-код. Постройте график зависимости числа коллизий от кол-ва экземпляров.
- ☐ (25 б.) Аналогично предыдущей задаче исследуйте еще 9 хэш-функций из этой [статьи](#). Данные хэш-функции предназначены для строк, поэтому вам потребуется написать генератор уникальных случайных строк. Рекомендую ограничиться английским алфавитом в нижнем регистре – из 6 букв можно составить 308915776 уникальных строк. Для проверки своих результатов можете посмотреть некоторые результаты в этой [статье](#).
- ☐ (25 б.) Вам необходимо написать контейнер для хранения записей телефонного справочника, однако разные клиенты собираются использовать его по-разному. Городская типография собирается напечатать справочник и ей нужны записи в отсортированном по фамилии человека порядке. Рекламное агентство нуждается в произвольном доступе к записям справочника. Регулярный пользователь хочет за максимально короткое время находить нужную ему запись. Удовлетворите желания клиентов, создав контейнер на базе Boost.Multiindex. Напишите демонстрационный код с использованием всех интерфейсов этого контейнера.