

Вопросы:

- В каких ситуациях используются контейнеры типа множества и отображения?
Когда мы работаем с отсортированными данными. Удобны операции вставки, удаления и поиска, которые работают за $O(\log N)$
- Каким требованиям должна удовлетворять качественная хеш функция? Хеш функция должна быть 1)детерминирован 2)равномерность 3)быстрота вычисления(не зависит от длины входа)
- Из-за чего в хеше возникают коллизии и как их разрешить? Количество коллизий зависит от хеш функции (ее метода задания) методы разрешения: 1)свести все к *forwardlist* 2)использовать открытую адресацию
- Почему сложность основных операций хэш-таблиц в худшем случае $O(N)$? 1)Заполняя таблицу происходит перевыделение памяти и перехэширование всех данных 2)Если *forwardlist* , то не менее $O(N)$
- В чём заключается преимущество интерфейсов контейнера `boost.Multiindex`? Можно работать с одними данными эффективно. Т.к. идет работа с набором ссылок на исходные данные.