1) Когда используются контейнеры типа (мульти) множество и отображение?

Контрейнеры типа (мульти) множество и отображение используются для хранения данных в отсортированном (по ключу в случае отображения) виде (мульти допускает хранение равных с точки зрения дерева элементов).

2) Каким требованиям должна удовлетворять качественная хэш-функция?

1) Детерминированность (всегда одинаковый хэш для одинаковых входных данных)
2) Скорость вычисления хэша для одного объекта не зависит от числа добавленных в хэш-таблицу элементов, но зависит от природы этого объекта (насколько сложен процесс его хэширования).
3) Равномерность (из равномерности поступающих объектов следует равномерность заполнения хэш-таблицы).

3) Из-за чего в хэш-функциях возникают коллизии и как можно их разрешать?

Коллизии возникают тогда, кгода несколько добавленных объектов имеют хэши, соответствующие одной ячейке. Разрешаются либо методом цепочек (на месте ячеек создается список из всех попадающих туда элементов), либо методом открытой адресации (элементы попадают в первую свободную ячейку, идущую за вычисленной).

4) Почему сложность основных операций хэш-таблиц в худшем случае O(N)?

При массовой коллизии получится либо один общий список, либо линейная хэш-таблица без каких либо ассоциаций между хэшем и позицией в списке, а тогда обходить придется весь массив из N элементов, как в обычном случае.

5) Что позволяет сделать инструмент создания контейнеров Boost.Multiindex?

Boost.Multiindex позволяет создавать разные структуры данных на основе одного набора элементов без их копирования (с использованием только их исходных номеров, что актуально для объемных типов).