

Контрольные вопросы №3

Аляев Роман

Когда используются контейнеры типа (мульти) множество и отображение?

Тип множество – `set` – упорядоченное множество уникальных ключей. Он нужен для того, чтобы хранить все элементы в отсортированном виде, даже при добавлении новых элементов.

Тип отображение – `map` – реализует ассоциативный массив, хранящий пары. Его можно использовать, когда каждому уникальному ключу мы хотим сопоставить несколько значений (например, записная книжка).

Каким требованиям должна удовлетворять качественная хэш-функция?

Хэш-функция удовлетворяет следующим трем требованиям:

- Быстро работает
- Детерминированность
- Равномерность

Из-за чего в хэш-таблицах возникают коллизии и как можно их разрешать?

Коллизии происходят, когда для данной задачи получается слишком мало остатков деления хэша на некоторое число h (т.е. это число подобрано неудачно), и в ячейку хэш-таблицы будет попытка записать два элемента. Существует два метода решения подобной проблемы:

Метод цепочек – из каждой ячейки делаем список (`forward_list` или `list`), который может расширяться сколько угодно

Метод открытой адресации – новый элемент, если его место занято, спускается вниз по таблице до тех пор, пока не найдет новую свободную ячейку

Почему сложность основных операций в хэш-таблицах в худшем случае $O(N)$?

Сложность основных операций может быть в худшем случае $O(N)$, так как:

Слишком много данных, если плохо создать хэш-функцию, могут попасть в одну ячейку и внутри нее поиск будет проходить дольше

Если хэш-таблица достигла критического порога заполнения, то ей придется перехэшироваться, а для этого нужно будет пересчитать заново все остатки элементов в таблице [N штук].

Что позволяет сделать инструмент создания контейнеров `Boost.Multiindex`?

`Boost.Multiindex` позволяет создать разные интерфейсы (в зависимости от нашей задачи) для работы с одним и тем же хранилищем данных. К примеру, есть какой-то массив данных: одному клиенту нужно максимально быстро найти какую-либо информацию

оттуда [hash]; второму клиенту нужен доступ по индексу в этом массиве [vector]; третьему нужны эти же данные в отсортированном порядке [set].