**13. Стандартные контейнерные типы. Ассоциативные контейнеры. Оценка сложности основных операций.**

**Стандартные последовательные контейнерные типы –** структуры данных, представляющие собой ассоциативные контейнеры для хранения данных в виде пар "ключ-значение".

**Виды:**

* **Словарь(std::map) #include <map>**
  + Набор уникальных пар: ключ-значение(pair<int, int>)
  + Хранится в виде бинарного дерева – способ упорядочивания последовательности значений, когда дерево узлов, каждый из узлов может иметь по два потомка, где справа те потомки, которые больше родителя, а слева - меньше. Где эти узлы представляют собой ключи пары: ключ-значение. Что ускоряет скорость поиска. Хранится в упорядоченном по ключе виде.
  + **Multimap** 
    - Как map, только можно добавлять дублирующиеся ключи.
  + **Unordered\_map**
    - Изображение выглядит как текст, снимок экрана, Шрифт

      Автоматически созданное описаниеКак map, только не отсортированный в виде бинарного дерева.
* **Множество(std::set) #include <set>**
  + Набор уникальных элементов.
  + Хранится в виде бинарного . то ускоряет скорость поиска. Хранится в упорядоченном по значению.
  + **Multiset.**
    - Как обычный set, но позволяющий добавлять повторяющиеся элементы.
  + **Unodered\_set: #include <unordered\_set>**
    - Изображение выглядит как текст, снимок экрана, Шрифт

      Автоматически созданное описаниеСтруктура уникальных значений, но не отсортированных бинарным деревом.

**Сложность:**

* **Map:** 
  + Добавление/Удаление: O(logn)
  + Поиск(доступ по индексу): O(logn) – а в худшем случае O(n)
  + **Unordered\_map:** 
    - Добавление/Удаление: O(1) – а в худшем случае O(n)
    - Поиск(доступ по индексу): O(n)
* **Set:**
  + Добавление/Удаление: O(logn)
  + Поиск(доступ по индексу): O(logn) – а в худшем случае O(n)
  + **Unordered\_set**
    - Добавление/Удаление: O(1) – а в худшем случае O(n)
    - Поиск(доступ по индексу): O(n)