Hash Table vs STL M

This article focus on : Compare and contrast Hash table and an STL Map. How is the hash table implemented? If the number of inputs is small, which data structure options can be used instead of a hash table?

Hash table

In a hash table, a value is stored by calling a hash function on a key.

- Values are not stored in sorted order.
- Additionally, since hash tables use the key to find the index that will store the value, an insert or lookup can be done in amortised O(1) time (assuming few collisions in the hash table).
- In a hash table, one must also handle potential collisions. This is often done by chaining, which means to create a linked list of all the values whose keys map to a particular index.

Implementation of Hash Table : A hash table is traditionally implemented with an array of linked lists. When we want to insert a key/Value pair, we map the key to an index in the array using the hash function. The value is then inserted into the linked list at that position.
Note: The elements in the linked list at a particular index of the array do not have the same key. Rather, hash function(key) is the same for these values. Therefore, in order to retrieve the value for a specific key,we need to store in each node both the exact key and the value.

To summarize, a hash table will be implemented with an array of linked lists, where each node in the linked list holds two pieces of data: the value and the original key. In addition, we will want to note the following design criteria:

1. We want to use a good hash function to ensure that the keys are well distributed. If they are not well distributed, then we would get a lot of collision and the speed to find an element would decline.
2. No matter how good hash function is, we will still have collisions, so we need a method for handling them. this often means chaining via a linked list, but it's not the only way.
3. We may also wish to implement methods to dynamically increase or decrease the hash table size depending on capacity. For example, when the ratio of the number of elements to the table size exceeds a certain threshold, we may wish to increase the hash table size. This would mean creating a new hash table and transferring the entries from the old table to the new table. Because this is an expensive operation, we want to be careful to not do it too often.

STL Map

The container map is an associative container included in the standard library of C++. The definition of this class is in the header file "map" of the namespace std.

STL Map Internal Implementation:
It's implemented as a self-balancing red-black tree. Probably the two most common self balancing trees are red-black tree and AVL trees. To balance the tree after an insertion/update both algorithms use the notion of rotations where the nodes of the tree are rotated to perform the re-balancing. While in both algorithms the insert/delete operations are O(log n), in the case of Red-Black tree re-balancing rotation is an O(1) operation while with AVL this is a O(log n) operation, making the RB tree more efficient in this aspect of the re-balancing sage and one of the possible reasons that is more commonly used.

Differences between hash table and STL map

1. **Null Keys :** STL Map allows one null key and multiple null values whereas hash table doesn't allow any null key or value.
2. **Thread synchronization :** Map is generally preferred over hash table if thread synchronization is not needed. Hash table is synchronized.
3. **Thread safe:** STL Maps are not thread safe whereas Hashmaps are thread safe and can be shared with many threads.
4. **Value Order :** In STL map, values are stored in sorted order whereas in hash table values are not stored in sorted order
5. **Searching Time :** You can use STL Map or binary tree for smaller data(Although it takes O(log n) time, the number of inputs may be small enough to make this time negligible) and for large amount of data, hash table is preferred.

Related articles

- Advantages of BST over hashmap
- Differences between HashMap and HashTable in Java

This article is contributed by **Brahmani Sai**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Recommended Posts:

Advantages of BST over Hash Table
Implementing our Own Hash Table with Separate Chaining in Java
Implementing own Hash Table with Open Addressing Linear Probing in C++
What are Hash Functions and How to choose a good Hash Function?
Difference between Fact Table and Dimension Table
MD5 hash in Java
std::hash class in C++ STL
Graph representations using set and hash
Hash Function for String data in C#
Sorting using trivial hash function
Sort elements by frequency | Set 4 (Efficient approach using hash)
Overview of Data Structures | Set 2 (Binary Tree, BST, Heap and Hash)
Direct Address Table
Find two Fibonacci numbers whose sum can be represented as N

Article Tags : Difference Between Hash HashTable STL

Practice Tags : Hash STL

9

2.8

To-do Done

Based on **9** vote(s)

Feedback/ Suggest Improvement | Add Notes | Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments