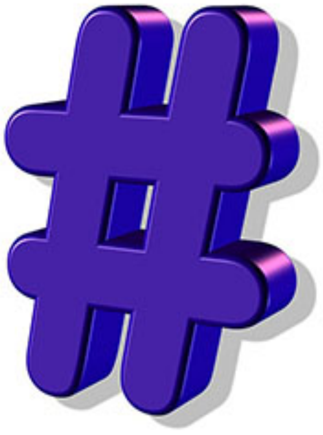
Data Structure in C - HashMap

Posted on Nov. 10, 2017



In this part of Data Structure in C tutorial series, I will explain what is HashMap and I will show you - how to write a HashMap from scratch in C. HashMap is a very important data structure and is often asked in interviews. Candidates are asked to explain and write complete HashMap code from scratch

Kaushik Baruah

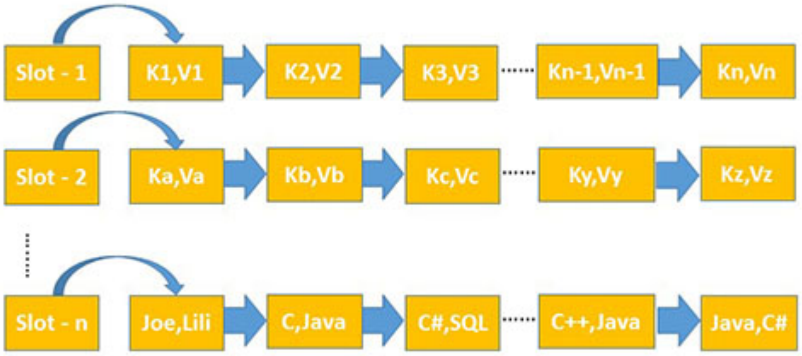
Follow @kauarba (https://twitter.com/kauaraba

ABOUT

My name is Kaushik Baruah and I am the chief blogger on this Blog. I have developed this Blog from scratch using Django as the backend and here I like to share my experience as software engineer and research engineer with my online readers. I will try to focus on caree planning, latest emerging technologies and tutorials on various computer scienc subjects. You can follow me on Twitter (https://twitter.com/kauaraba), Facebook (https://www.facebook.com/kaushik.baruah.92) and Google+ (https://plus.google.com/u/0/104727781243061209730/posts)

POPULAR POSTS

A HashMap looks like as shown below



HashMap data structure contains Key-Value pairs. All the keys in a HashMap data structure are unique. The values are not necessarily be unique. Different keys may correspond to same value. You can see in the above diagram that all the Keys are unique while some values are duplicated.

How HashMap Works

Suppose you are asked to prepare a spouse database. You are given a list of Husbands and Wives where names of the Husbands are unique. You need to store them in such a way that - if someone gives you the name of the Husband, you need to get the name of the Wife.

You can then think of Husband and Wife as key value pair. Since the Husband names are unique, you can use them as keys and names of the Wives as values.

You can then store these key value pairs as shown in the above table.

How to Insert a Key-Value pair in the HashMap data structure

You have to Insert the Key-Value pairs keeping your ultimate goal in mind. Your ultimate goal is to return the name of the wife, given the Husband name. How fast you can do that ?

If you simply insert the Key-Value pairs in a simple list, you have to scan the entire list to find the Husband

(http://www.kaushikbaruah.com/posts/6-tips-to-become-android-developer)
6 Tips to become an Android Developer (http://www.kaushikbaruah.com/posts/6-tips-to-become-android-developer)

(http://www.kaushikbaruah.com/posts/xerox-research-centre-india-interview-experience)
Xerox Research Interview Experience (http://www.kaushikbaruah.com/posts/xerox-research-centre-india-interview-experience)

name so that you can return the corresponding wife name. With HashMap we will try to reduce the search area.

We will try to break this long list into **n** small lists as shown in the figure above. Whenever a Husband name will be given - we will look for the name in one of these small lists. If the name is found, we return the Wife name. Otherwise, there is no Husband-Wife pair for the given Husband name.

To break the long list into these **n** small lists, we will need something called a Hash Function (https://en.wikipedia.org/wiki/Hash_function).

While Inserting a Husband-Wife pair into the HashMap data structure, we will give the Husband name as input to this Hash function and the Hash function will return us a slot number which the number of the list in which we will insert the Key-Value or the Husband-Wife pair.

As shown in the figure above - slot-1, slot-2, ... , slot-n represents **n** lists and we say that size of the HashMap is **n**.

The main trick here is to chose your Hash Function (https://en.wikipedia.org/wiki/Hash_function) cleverly. If our HashMap size is 5 and we have 10 pairs to insert, in the ideal case we will try to have 2 pairs to be inserted in each of the 5 lists so that when we will search for a Husband name which may map to any of these 5 lists, we only have to search in a list of length 2.

If the choice of the Hash Function (https://en.wikipedia.org/wiki/Hash_function) is bad, then in the worst case, the Hash Function (https://en.wikipedia.org/wiki/Hash_function) may return the same slot number for all the 10 pairs and all of the 10 pairs may end up in the same list. Therefore, in the worst case - to look for a Husband name we may have to scan all the 10 pairs.

Therefore, define your Hash function cleverly. There are various methods on How to chose an optimal Hash

(http://www.kaushikbaruah.com/posts/top-5-tips-to-clear-hr-interview)
Tips to clear the HR Interview (http://www.kaushikbaruah.com/posts/top-5-tips-to-clear-hr-interview)

(http://www.kaushikbaruah.com/posts/data-structure-in-c-hashmap)
HashMap data structure in C (http://www.kaushikbaruah.com/posts/data-structure-in-c-hashmap)

(http://www.kaushikbaruah.com/posts/career-paths-for-fresh-graduate)
Career paths for fresh Graduates (http://www.kaushikbaruah.com/posts/career-paths-

Function (https://en.wikipedia.org/wiki/Hash_function). Double Hashing (https://en.wikipedia.org/wiki/Double_hashing) , Universal Hashing (https://en.wikipedia.org/wiki/Universal_hashing), Perfect Hashing (https://en.wikipedia.org/wiki/Perfect_hash_function) to name a few. You can go through these techniques in detail.

How to look for a Key in the HashMap

To look for a Key-Value pair, we use the same technique as described above. We give the Key as input to the Hash function and get the slot number or the List number in which we need to search for that Key.

We then search for the Key in the corresponding List. If we find the Key in the List, then we return the corresponding value for that Key.

Below is the complete working code for HashMap data structure in C. I have taken Keys and Values as integer. You can modify the code to change the data types of Keys and Values as per your requirement.

for-fresh-graduate)

(http://www.kaushikbaruah.com/posts/7-tips-to-improve-your-coding-skill)

7 Tips to improve Coding Skill (http://www.kaushikbaruah.com/posts/7-tips-to-improve-your-coding-skill)

```
#include <stdio.h>
#include <stdlib.h>
struct node{
    int key;
    int val;
    struct node *next;
};
struct table{
    int size;
    struct node **list;
};
struct table *createTable(int size){
    struct table *t = (struct table*)malloc(sizeof(stru
    t->size = size;
    t->list = (struct node**)malloc(sizeof(struct node*
    int i;
    for(i=0;i<size;i++)
        t->list[i] = NULL;
    return t;
}
int hashCode(struct table *t,int key){
    if(key<0)
        return -(key%t->size);
    return key%t->size;
}
void insert(struct table *t,int key,int val){
    int pos = hashCode(t,key);
    struct node *list = t->list[pos];
    struct node *newNode = (struct node*)malloc(sizeof(
    struct node *temp = list;
    while(temp){
        if(temp->key==key){
            temp->val = val;
            return;
        }
        temp = temp->next;
    }
    newNode->key = key;
    newNode->val = val;
    newNode->next = list;
    t->list[pos] = newNode;
}
int lookup(struct table *t,int key){
    int pos = hashCode(t,key);
    struct node *list = t->list[pos];
    struct node *temp = list;
    while(temp){
        if(temp->key==key){
            return temp->val;
        }
        temp = temp->next;
    }
    return -1;
}
int main(){
    struct table *t = createTable(5);
    insert(t,2,3);
    insert(t,5,4);
    printf("%d",lookup(t,5));
    return 0;
}
```

If you have any doubt regarding the code or any other concept, please leave your queries in the comment section.

SUBSCRIBE

RECOMMENDED POSTS FOR YOU

(http://www.kaushikbaruah.com /posts/java-hashmap/)

Generic Java HashMap from Scratch (http://www.kaushikbaruah.com /posts/java-hashmap/)

(http://www.kaushikbaruah.com /posts/7-tips-to-improve-your-coding-skill)

7 tips to improve Coding Skill (http://www.kaushikbaruah.com /posts/7-tips-to-improve-your-coding-skill)

(http://www.kaushikbaruah.com /posts/data-structures-in-c-stack/)

Data Structures in C - Stack (http://www.kaushikbaruah.com /posts/data-structures-in-c-stack/)

About Us

My name is Kaushik Baruah and I am the chief blogger on this Blog. I have developed this Blog from scratch using Django as the backend and here I like to share my experience as software engineer and research engineer with my online readers. I will try to focus on career planning, latest emerging technologies and tutorials on various computer science subjects.

Get Free Updates

Your email Id...

SUBSCRIBE

Follow Us

Twitter (https://twitter.com /kauaraba)
Facebook (https://www.facebook.com /kaushik.baruah.92)
Google+ (https://plus.google.com /u/0/104727781243061209730 /posts)