

Data Structure 2018

Lab 07

MohammadSadegh Najafi

Oct 18, 2018.

Hashing

Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects

- In universities, each student is assigned a unique roll number that can be used to retrieve information about them.
- In libraries, each book is assigned a unique number that can be used to determine information about the book, such as its exact position in the library or the users it has been issued to etc

In both these examples the students and books were hashed to a unique number.

Hash function

- A hash function is any function that can be used to map a data set of an arbitrary size to a data set of a fixed size.
- The values returned by a hash function are called hash values
- To achieve a good hashing mechanism, It is important to have a good hash function
- Hash Function requirements:
 - Easy to compute: It should be easy to compute and must not become an algorithm in itself.
 - Uniform distribution: It should provide a uniform distribution across the hash table and should not result in clustering.
 - Less collisions: Collisions occur when pairs of elements are mapped to the same hash value. These should be avoided.

Hash example

Store these string in the hash table :

{“abcdef”, “bcdefa”, “cdefab” , “defabc” }

hash function:

The index for a specific string will be equal to the sum of the ASCII values of the characters modulo 599.

Hash example

Hash Table

Here all strings are sorted at same index

Index				
0				
1				
2	abcdef	bcdefa	cdefab	defabc
3				
4				
-				
-				
-				
-				

Hash example

Let's try a different hash function:

The index for a specific string will be equal to sum of ASCII values of characters multiplied by their respective order in the string after which it is modulo with 2069 (prime number).

String	Hash function	Index
abcdef	$(971 + 982 + 993 + 1004 + 1015 + 1026)\%2069$	38
bcdefa	$(981 + 992 + 1003 + 1014 + 1025 + 976)\%2069$	23
cdefab	$(991 + 1002 + 1013 + 1024 + 975 + 986)\%2069$	14
defabc	$(1001 + 1012 + 1023 + 974 + 985 + 996)\%2069$	11

Hash example

Hash Table

Here all strings are stored at different indices

Index	
0	
1	
-	
-	
-	
11	defabc
12	
13	
14	cdefab
-	
-	
-	
-	
23	bcdefa
-	
-	
-	
38	abcdef
-	
-	

Today's Task

1- Queue using Stacks

We are given a stack data structure with push and pop operations, the task is to implement a queue using instances of stack data structure and operations on them.

A queue can be implemented using two stacks. Let queue to be implemented be q and stacks used to implement q be $stack1$ and $stack2$.

Implementation Methode: This method makes sure that oldest entered element is always at the top of stack 1, so that deQueue operation just pops from stack1. To put the element at top of stack1, stack2 is used.