CSP2348 Data Structures

Solutions to Tutorial 07: Hash Table Data Structures

(For reference only)

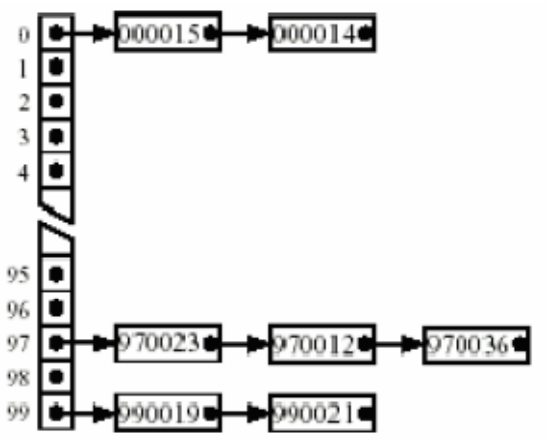
Tasks:

Complete the following.

Task 1: Attempt Exercise 12.1 given on page 334 in the textbook.

a. Notice the insertion process in a CBHT;
b. Discuss whether the given hash function is a right choice or not.

(a)

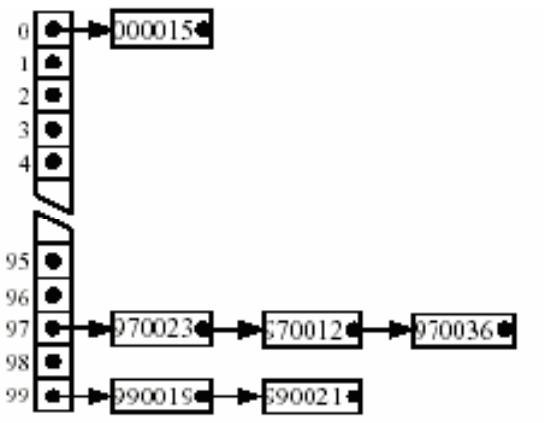


(b) The numbers of comparisons when the CBHT is searched for each key are:

Id	000014	990021	990019	970036	000015	970012	970023
No of comprsn	2	2	1	3	1	2	1

The average number of comparisons is (2+2+1+3+1+2+1)/7 ≈ 1.7.

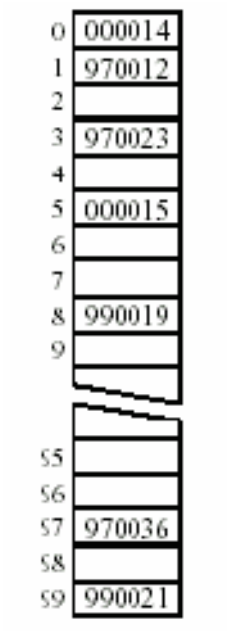
(c)



Task 2: Attempt Exercise 12.2 given on page 334 in the textbook.

a. Notice the insertion process in a OBHT;
b. Discuss whether the given hash function is a right choice or not.

(a)



(b)

The number of comparisons when the OBHT is searched for each key is:

000014	990021	990019	970036	000015	970012	970023
1	1	2	1	2	3	3

The average number of comparisons is (1+1+2+1+2+3+3)/7 ≈ 1.9.

(c)

Task 3: Test the Java program WS0601 (Download the Java code from Blackboard).

a. Analyse the hash function given in the program;
b. Execute this program;
c. Draw sketches of OBHT and CBHT to show the executed result. Draw the search path as well if any collision occurs.

(a)

The hash function values are computed by the statement

```
return (object.hashCode() & MASK) % CAPACITY;
```

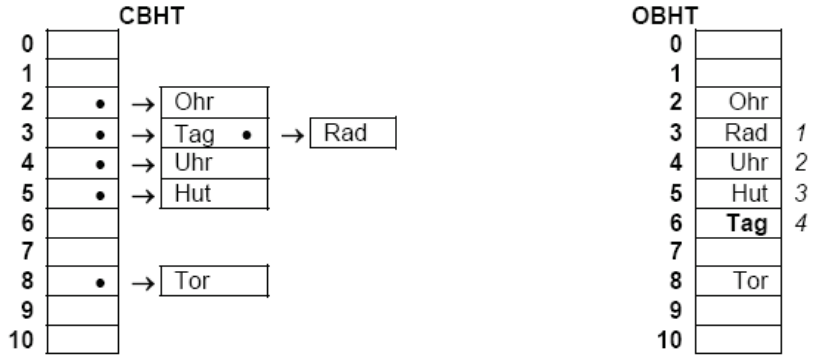
where `CAPACITY` is 11 and `MASK` is 2147483647 expressed in hexadecimal as 0x7FFFFFFF. The operation n & MASK simply removes the

sign from whatever integer n has. As such, the resulting value returned by the hash() function is guaranteed to be in the range 0 to 10.

(b) *executed result:*

```
hash(Rad) = 3
hash(Uhr) = 4
hash(Ohr) = 2
hash(Tor) = 8
hash(Hut) = 5
hash(Tag) = 3
```

(c)



Task 4: Test the Java program WS0602 (Download the Java code from Blackboard).

a. Execute this program;
b. Discuss why there is no collision occurred in this program.

(a) executed result:

```
hash(Rad) = 99
hash(Uhr) = 82
hash(Ohr) = 73
hash(Tor) = 45
hash(Hut) = 13
hash(Tag) = 4
```

(b)

The CAPACITY, and thus the bucket number, has been increased to 101 that is a prime number. This reduces the possibility of collision occurrence.