

W6 hash

2019年2月28日 星期四 下午1:49

1. In java , the hash code for String values is computed as:

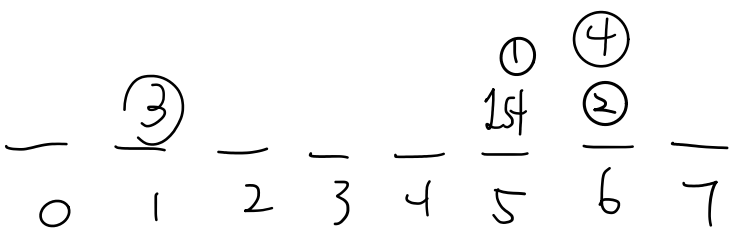
Ans: $C_0 \times 31^{N-1} + C_1 \times 31^{N-2} + \dots + C_{N-2} \times 31^1 + C_{N-1} \times 31^0$
* C_i is the ASCII value for character at position i
and n is number of characters in the string

How about Double?

Ans: $(int)(bits \wedge (bits >> 32))$

2. What are the first 6 indexes of the probe sequence for a key with a hash index 5 on hash table with 8 elements that uses quadratic probing?

* P.S = $H_k, H_k + 1^2, H_k + 2^2, \dots$



$5, 5+1^2, 5+2^2, 5+3^2, \dots$
6 9 14

3. linear probing

Que - 3. A hash table of length 10 uses open addressing with hash function $h(k)=k \bmod 10$, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

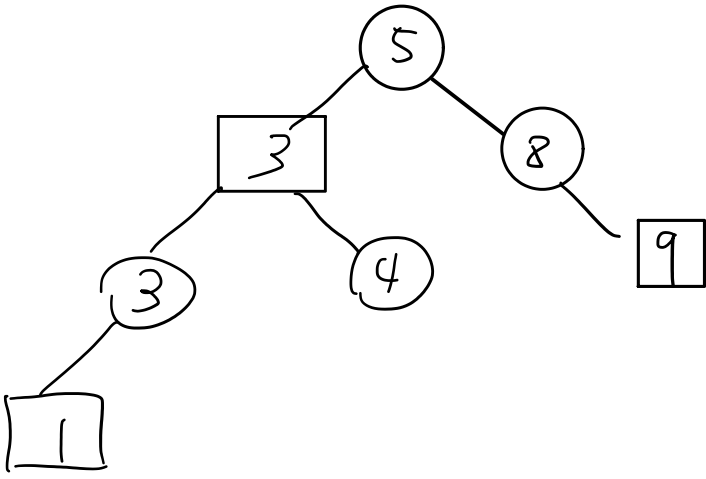
(A) 46, 42, 34, 52, 23, 33
(B) 34, 42, 23, 52, 33, 46
(C) 46, 34, 42, 23, 52, 33
(D) 42, 46, 33, 23, 34, 52

Solution: We will check whether sequence given in option A can lead to hash table given in question. Option A inserts 46, 42, 34, 52, 23, 33 as:

For key 46, $h(46)$ is $46 \% 10 = 6$. Therefore, 46 is placed at 6th index in the hash table.
For key 42, $h(42)$ is $42 \% 10 = 2$. Therefore, 42 is placed at 2nd index in the hash table.
For key 34, $h(34)$ is $34 \% 10 = 4$. Therefore, 34 is placed at 4th index in the hash table.
For key 52, $h(52)$ is $52 \% 10 = 2$. However, index 2 is occupied with 42. Therefore, 52 is placed at 3rd index in the hash table. But in given hash table, 52 is placed at 5th index. Therefore, sequence in option A can't generate hash table given in question.
In the similar way, we can check for other options as well which leads to answer as (C).

4. R-B-T ☐ black ☐ Red

5, 8, 2, 9, 4, 3, 1



(Try more on visualization web)

1. Is the following tree height balanced, full, complete?

2. Insert 10, 27, 40, 19, 6, 20. in an AVL Tree.

3. 2-3 tree: insert 3, 4, 5, 1, 2, 9, 7, 10

1) delete 5: borrow from sibling

2) delete 4: using suc/pre

delete 7: merge & demote from parent.

3-4 tree: delete 7, delete 4.

A.

5. Hashing: traversal $O(T.S+N)$ if bucket
 $O(T.S)$ if open addr