CSCI 321 Computer Science III Fall 2018

Lecture 5 Activity 1

1. Draw the 11-entry hash table that results from using the hash function, h(i) = (3i+5) mod 11, to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, assuming collisions are handled by chaining.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 13 | 94 |  |  |  | 44 |  |  | 12 | 16 | 20 |
|  | 39 |  |  |  | 88 |  |  | 23 | 5 |  |
|  |  |  |  |  | 11 |  |  |  |  |  |

I wrote a quick python script to calculate these values:

hashTable = [12,44,13,88,23,94,11,39,20,16,5]

def hashEqn(x):

return ((3 \* x) + 5) % 11

for i in hashTable:

print(i, hashEqn(i))

1. Redo P1 assuming collisions are handled by linear probing.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 13 | 94 | 39 | 16 | 5 | 44 | 88 | 11 | 12 | 23 | 20 |

1. Redo P1 assuming collisions are handled by double hashing using the secondary hash function d(k) = 7− (k mod 7).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | 13 | 12 | 88 | 94 | 44 |  |  |  |  |  |
|  | 20 | 5 | 11 |  | 23 |  |  |  |  |  |
|  |  |  | 39 |  | 16 |  |  |  |  |  |

I wrote a quick python script to compute these values for the double hash:

hashTable = [12,44,13,88,23,94,11,39,20,16,5]

# def hashEqn(x):

# return ((3 \* x) + 5) % 11

def doubleHash(x):

return((7-x) % 7)

print("KEY\t|\tDoubleHash")

for i in hashTable:

print(i ,'\t|\t',doubleHash(i))

1. Hash code is used to “encode” general keys into integers. One approach of creating a hash code is to use Java's hashCode() method. The hashCode() method is implemented in the Object class and therefore each class in Java inherits it. The hash code provides a numeric representation of an object (this is somewhat similar to the toString method that gives a text representation of an object).
2. Write a program to show the hash code of the following.

Integer 2018,

String “2018”,

StringBuffer “2018”,

ArrayList with first element is Integer 2018.

import java.util.ArrayList;

public class lec5a1 {

public static void main(String[] args) {

Integer hashInt = 2018;

String hashString = "2018";

StringBuffer hashSB = new StringBuffer(2018);

ArrayList<Integer> hashArrInt = new ArrayList<Integer>(2018);

System.out.println(hashInt.hashCode());

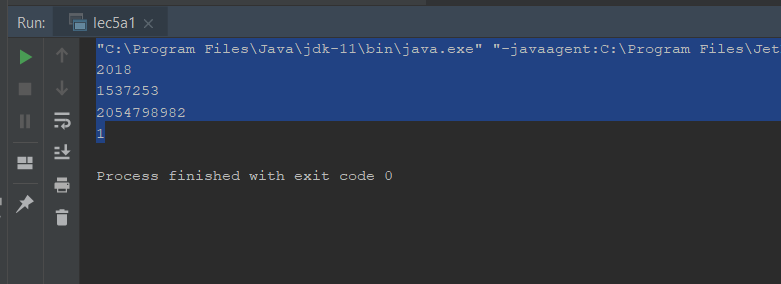
System.out.println(hashString.hashCode());

System.out.println(hashSB.hashCode());

System.out.println(hashArrInt.hashCode());

}

}

OUTPUT: 

1. Verify that different objects might have the same hashcode. For example, show the hashcode for the strings “Aa” and “BB”.

public class lec5a1 {

public static void main(String[] args) {

String stringA = "Aa";

String stringB = "BB";

System.out.println(stringA.hashCode());

System.out.println(stringB.hashCode());

}

}

