KTU NOTES

The learning companion.

KTU STUDY MATERIALS | SYLLABUS | LIVE NOTIFICATIONS | SOLVED QUESTION PAPERS

DATA STRUCTURES –ITT 201 (S3 IT)

MODULE - 5

	Module 5: Hash Table	7 hrs
5.1	**Hash Tables**-Hash Functions- Features of hash function.	1
5.2	**Different Hash Functions:** Division Method- Multiplication Method - Mid Square Method, Folding Method- related problems.	2
5.3	**Collision Resolution Techniques:** Closed hashing (Linear probing) and Open Hashing (Separate Chaining) . Closed hashing(Linear probing) -Drawbacks- Remedies - Radom Probing – Double hashing/Re-hashing –Quadratic Probing, problems to create hash tables using linear probing and Random probing, double hash and quadratic probing .	3
5.4	Open Hashing (Separate Chaining)	1

MODULE 5

HASH TABLE

HASH TABLES:

• Hash Table is a data structure which stores data in an associative manner.

• In a hash table, data is stored in an array format, where each data value has its own unique index value. Access of data becomes very fast if we know the index of the desired data.

• Hashing function is used to implement hash table.

• Hash function returns a location in hash table.

HASHING FUNCTIONS:

A **hash function** is any function that can be used to map data of arbitrary size to fixed-size values. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes. The values are usually used to index a fixed-size table called a hash table. **Or** it can also be said as follows: A **hash function** is a function that converts a given numeric or alphanumeric key to a small practical integer value. The mapped integer value is used as an index in the hash table.

The main features of the hashing function is:

- No two input hashes should map to the same output hash.
- It should be difficult to guess the input value for a hash function from its output.

Some of the main hashing functions are:

1. Mid Square method
2. Division method
3. Folding method
4. Multiplication method

Mid Square Method:

Mid-Square hashing is a hashing technique in which unique keys are generated. In this technique, a key value is taken and it is squared. Then, some digits from the middle are extracted. These extracted digits form a number which is taken as the hash value.

h(k)=mid(k^2)

For example;

Assume the key is 12.

Then the key is squared. h(k)=144

That is the hash value=4. That is the key 12 is assigned to the hash table at the index 4.

[For more details: https://youtu.be/GJ13Ks33GzU]

Division Method:

The division method involves mapping a key k into one of m slots by taking the remainder of k divided by m as expressed in the hash function.

h(k) = k mod m .

(m is the size of the hash table.)

For example;

Assume the key is 12 and the size of the hash table m is 6.

h(k)= 12 mod 6

=0

That is the hash value=0. That is the key 12 is assigned to the hash table at the index 0.

[For more details: https://youtu.be/GJ13Ks33GzU]

Folding Method:

It breaks up a key value into precise segments that are added to form a hash value.

For example;

Assume the key is 456789123 and m is 1000.

Now the key value is separated into fragments :456/789/123.

They are then added together which gives a sum of 1368. Since the 1368 doesn't belongs to the hash table the folding method is again applied.

The key 1368 separated again into fragments: 136/8. The 136 and 8 is the added and we get a sum of 144. Since the obtained value is there in the hash table the key value 456789123 get assigned to the index 144 of the hash table.

[For more details: https://youtu.be/GJ13Ks33GzU]

Multiplication Method:

$$h(k) = \lfloor m(k\,A \pmod 1)) \rfloor$$

m is the size of the hash table.

k is the key value.

A is an irrational number randomly chosen between 0 and 1.

Also the hash value should only be the floor value of the answer.

For example;

Assume the key is 6 and m is 32.

And the A is 0.3(any random value between 0-1).

Now the key is multiplied with A. (6*0.3=1.8)

Then modulo 1 the answer. [(1.8) mod 1=0.018]

It is the multiplied with the m. [(0.018*32) =0.576]

The hash value should only be the floor value. So the hash value is 0. That is the key 6 is assigned to the index 0 of the hash table.

[For more details: https://youtu.be/Gd3C0-Bt2IA]

Collision Resolution Techniques:

Collision resolution techniques in hashing are used to resolve collisions in hashing.

Collision resolution techniques are:

1.Open Hashing or Separate Chaining or Closed Addressing

2.Closed Hashing or Open Addressing

Open Hashing:

Open hashing is a collision avoidance method which uses array of linked list to resolve the collision.

It is also known as the separate chaining method (each linked list is considered as a chain).

Here, we use additional memory space.

Keys are stored in linked lists attached to cells of a hash table.

The appropriate index value of the specific key is identified using any of the hashing functions (mostly division method) and then the keys are inserted to the corresponding value. If same index value for more than one key (collision) then it is stored in linked lists attached to the cells of the hash table.

For example:

Consider the following keys: 55,61,1,10,25

Let m be 5.

First of all we finds the corresponding hash values of the keys-

By using division method:

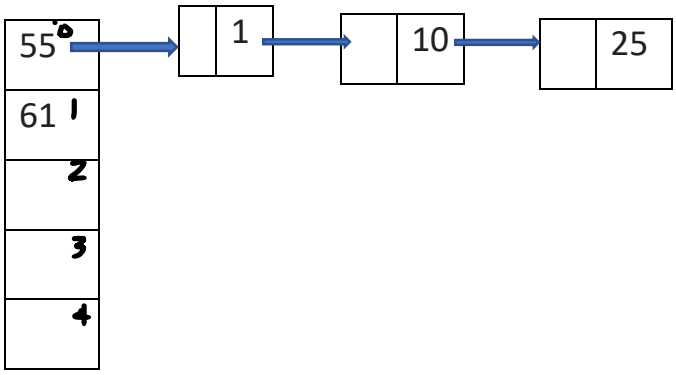
h(55)= 55 mode 5 = 0

h(61)= 61 mod 5 =1

h(1)= 1 mod 5 = 0

h(10)= 10 mod 5 = 0

h(25)= 25 mode 5 = 0



Here, the key values 55,1,10,25 have the same hash value. So they are stored in linked list attached to the cell of the hash table with the index 0. And the key 61 is stored in the cell of the hash table with index 1.

[For more details: https://youtu.be/Dk57JonwKNk]

Closed hashing:

If collision occurs when hashing performs, the values are transferred in to the alternative free location.

It is also called as Open Addressing.

Here, we do not use additional memory space.

There are mainly three types of open addressing:

1.Linear Probing

2.Quadratic Probing

3.Double Hashing

Linear Probing:

A hash table in which a collision is resolved by putting the item in the next empty place in the array following the occupied place.

In linear probing, the hash table is searched sequentially that starts from the original location of the hash.

If in case the location that we get is already occupied, then we check for the next location. (k➡ k+1➡k+2 ….)

The function used for rehashing is as follows: **h(k) = (k+i) mod m**.

For example;

Consider the key values: 10,21,44,18,31

Let the m be 5.

First of all we need to find the corresponding hash values for the keys-

By using the division method:

h(10)= 10 mod 5=0

h(21)=21 mod 5 =1

h(44)=44 mod 5 = 4

h(18)=18 mod 5=3

h(31)=31 mod 5=1

Here the cell with index 1 is already filled with the key 21. Therefore here the collision occurs. That is the value of i=1 from 0.

Now it goes for rehashing. h(31)=(31+1) mod 5=2.

Since the index 2 of the given hash table is not assigned the key value 31 get assigned to the index 2.

10
21
31
18
44

(The value of i increases according to the number of collisions occurs for a particular element.)

[Number of searches=n]

Drawback:

Primary Clustering: One of the problems with linear probing is Primary clustering, many consecutive elements form groups and it starts taking time to find a free slot or to search for an element.

Quadratic Probing:

A method of open addressing for hash table in which a collision is resolved by putting the item in the next empty place given by a probe sequence.

The space between places in the sequence increases quadratically.

Quadratic probing is an open addressing technique for resolving hash collisions in hash tables. Quadratic probing operates by taking the original hash

index and adding successive values of an arbitrary quadratic polynomial until an open slot is found.(k ➡ k+1^2 ➡ k+2^2 ……..)

The function used for rehashing is as follows: **h(k) = (k+i^2) mod m**.

For example:

Consider the key values: 55,30,12,7

Let the m be 5.

First of all we need to find the corresponding hash values for the keys-

By using the division method:

h(55)= 55 mod 5=0

h(30)=30 mod 5 =0

Here the cell with index 0 is already filled with the key 55. Therefore here the collision occurs. That is the value of i=1 from 0.

Now it goes for rehashing. h(30)=(30+1) mod 5=1.

Since the index 1 of the given hash table is not assigned the key value 30 get assigned to the index 1.

h(12)=12 mod 5 = 2

h(7)=7 mod 5=2

Here the cell with index 2 is already filled with the key 12. Therefore here the collision occurs. That is the value of i=1 from 0.

Now it goes for rehashing. h(7)=(7+1) mod 5=3.

Since the index 3 of the given hash table is not assigned the key value 7 get assigned to the index 3.

55
30
12
7

(The value of i increases according to the number of collisions occurs for a particular element.)

Drawback:

Its probe sequence typically will not visit all slots in the hash table.

It gives rise to secondary clustering.

Double Hashing:

Double hashing is a collision-resolving technique in open addressing hash tables. Double hashing uses the idea of applying a second hash function to the key when a collision occurs.

h(k)=h1(k)+[i*h2(k)]

For example:

Consider the key values: 18,41,22,44

Let the m be 13.

h1(k)=k mod 13

h2(k)=7-k mode 7

First of all we need to find the corresponding hash values for the keys-

Finding the hash values according to the functions given above:

Keys	h1	h2
18	5	3
41	2	1
22	9	6
44	5	5

h(18)=5+0*3=5

h(41)=2+0*1=2

h(22)=9+0*6=9

h(44)=5+0*5=5

Here the cell with index 5 is already filled with the key 18. Therefore here the collision occurs. That is the value of i=1 from 0.

Now it goes for rehashing. h(44)=5+1*5=10

Since the index 10 of the given hash table is not assigned the key value 44 get assigned to the index 10.

		41			18				22	44		

Drawbacks:

- Double hashing is more difficult to implement than any other.
- Double hashing can cause thrashing (Thrashing occurs when there are too many pages in memory, and each page refers to another page. The real memory shortens in capacity to have all the pages in it, so it uses 'virtual memory'.)

[For more details: https://youtu.be/zeMa9sg-VJM
https://youtu.be/dxrLtf-Fybk
https://youtu.be/AYcsTOeFVas]

Load factor:

Load factor=Number of keys/size of hash table

Difference between open hashing and closed hashing:

	Open Hashing	Closed Hashing
1.	No formation of cluster	Cluster formation
2.	Use of additional memory space	No use of additional memory space
3.	Have pointers	No pointers
4.	Deletion is easier	Deletion is hard
5.	No rearrangement is required	Rearrangement required
6.	Can have additional keys	Additional key is not possible
7.	Load factor between 0 and 1	Load factor is greater than 1

PREVIOUS YEARS QUESTIONS

ITT 201 DECEMBER 2019

1.What is the Hash function?

2.What are the general approaches to avoid the collision?

3.Define Hashing. Explain any four Hashing Techniques with example.

4. a) Explain the Separate Chaining technique.

b) What is Closed Hashing? Suppose the size of the Hash Table is 11. The hash function is H1 (K) = K(mod 11). Show how the key values K={16,12,27,23,8,41,13} stored in the Hash Table using the following techniques.

i) Linear Probing

ii) Double Hashing where H2 (K)=7 - (K mod 7)

CS 205 JANUARY 2017

1.Define hashing. What are the properties of a good hash function? With necessary examples explain four different hashing techniques.

2.Define collision. What is linear probing? The following keys 10, 16, 11, 1, 3, 4,23 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is the resultant hash table?

CS 205 JULY 2017

1.Explain mid-square method in hashing with an example.

2.What is hashing and what is its importance.

CS 205 DECEMBER 2017

1.a) Get the hash index in table of size 7 for the following list. 56,43,27,32,3.

b) Do the rehashing when the inserted elements are more than 4.

c) Briefly explain any 2 hasting functions.

CS 205 APRIL 2018

1.What is double hashing? Suppose size of the hash table is 11. Open addressing and double hashing is used to resolve collisions. The hash function used is H(k) = k mod 11. The second hash function is H2(k) = 5 - (k mod 5) What values will be in the hash table after the following sequence of insertions? 16, 23 9, 34, 12, 56.

CS 205 DECEMBER 2018

1.a) What are the characteristics of a good hash function? (4)

b) Demonstrate the insertion of the keys 5, 28, 15, 20, 33, 12, 17, 32 into a hash table with collisions resolved by linear probing. Let the table have 9 slots, with the starting index 0. Let the hash function be h(k) = k mod 9

2. a) What is Primary Clustering?

b) Given input keys {1, 3, 23, 9, 4, 29, 19} and a hash function h(X) = X mod table size. The initial hash table contains 10 slots, with starting index 0. Show the resulting table after rehashing when the load factor= 0.5, using linear probing.

CS 205 MAY 2019

1.a) What is meant by collision? Give an example. (2)

b) Explain the four different hashing functions with an example for each. (8)

2.Given the values {2341, 4234, 2839, 430, 22, 397, 3920} a hash table of size 7 and a hash function h(x) = x mod 7, show the resulting table after inserting the values in the given order with each of the following collision strategies.

(i) separate chaining

(ii) linear probing

(iii) double hashing with second hash function h1(x) = (2x – 1) mod 7.

CS 205 DECEMBER 2019

1.Where is a hash table data structure used? Explain any two commonly used hash functions with examples.

2.Let the size of a hash table be 7. The index of the hash table varies from 0 to 6. Consider the keys 89, 18, 49, 58, 25 in the order. Show how the keys are stored in the hash table (use linear probing).

3.Let the size of the hash table be 10. The index of the hash table varies from 0 to 9. Consider the keys 43, 24, 57, 12, 10, 64, 19, 82, 36, 39 in the order. Show how the keys are occupied. Use chaining method.

CS 205 SEPTEMBER 2020

1.What you mean by Open Addressing and Closed Addressing?

2.Define hashing, hash function and collision.

3.A hash table contains 7 buckets and uses linear probing to solve collision. The key values are integers and the hash function used is key%7. Draw the table that results after inserting in the given order the following values: 16,8,4,13,29,11,22.

4.Write short notes on separate chaining.

CS 205 DECEMBER 2020

1.How can linear probing be used to resolve collisions? Explain with example.

2.What are the limitations of linear probing? How can double hashing be used to resolve these limitations? Illustrate with examples.

3.What is open hashing? How is it used to resolve collisions in a hash table? Compare its performance with closed hashing.

4.Let the size of the hash table be 12. Consider the keys 43, 24, 57, 12, 10, 64,19, 82, 36, 39 in the order. Show how the keys are occupied using chaining method.

CS 205 JANUARY 2022

1.Define hashing and collision? Discuss the advantages and disadvantages of hashing over other searching techniques.

2. Consider a hash table with 9 slots. The hash function h(k) = k mod 9. The following keys are inserted in the order 5, 28, 19, 15, 20, 33, 12, 17, 10. Draw the contents of hash table when the collision are resolved by

1) Chaining

2) Linear Probing

3) Double hashing. The second hash function h2(x) = 7- (x mod 7)

3. a) What is hash function? Explain any two hash functions with examples.

b) Explain the different collision resolution techniques.