ECAP770

Advance Data Structures

Ashwani Kumar
Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- Understand Collision Resolution Techniques
- Open addressing (closed hashing)
  - Double hashing

Collision Resolution

- When two keys or hash values compete with a single hash table slot, then Collision occur.
- To resolve collision we use collision resolution techniques.
- Collisions can be reduced with a selection of a good hash function.

Open Addressing

- The open addressing technique requires a hash table with fixed and known size.
- All elements are stored in the hash table itself
- The size of the table must be greater than or equal to the total number of keys.
- During insertion, if a collision is encountered, alternative cells are tried until an empty bucket is found.

Open Addressing

In case of collision:

- Probing is performed until an empty bucket is found.
- Once an empty bucket is found, the key is inserted.
- Probing is performed in accordance with the technique used for open addressing.

Double Hashing

- Double Hashing is a hashing collision resolution technique in open addressed Hash tables.
- In double hashing, there are two hash functions.
- The second hash function is used to provide an offset value in case the first function causes a collision.
- Second hash function used to remove the collision when you encountered the collision.

Double Hashing

- Double hashing can be performed using :

  *(hash1(key) + i * hash2(key)) mod TABLE_SIZE*

  Here hash1() and hash2() are hash functions

- First hash function:

  hash1(key) = key mod TABLE_SIZE

- Second hash function is :

  hash2(key) = PRIME – (key mod PRIME)

where PRIME is a prime smaller than the TABLE_SIZE

Double Hashing

A good second Hash function is:

- It must never evaluate to zero
- Must make sure that all cells can be probed

Example: Double hashing

h1(k) = k mod 10

h2(key) = Prime – (key mod Prime)

h2(key) = 7 – (key mod 7)

(h1(key) + i * h2(key)) mod Table size

Elements: 20, 25, 36, 16, 55, 17

Index	Hash table
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Example: Double hashing

h1(k) = k mod 10

h2(key) = Prime – (key mod Prime)

(h1(key) + i * h2(key)) mod Table size

Elements: 20, 25, 36, 16, 55, 17

20 mod 10 = 0

Index	Hash table
0	20
1	
2	
3	
4	
5	
6	
7	
8	
9	

Example: Double hashing

h1(k) = k mod 10

h2(key) = Prime – (key mod Prime)

(h1(key) + i * h2(key)) mod Table size

Elements: 20, 25, 36, 16, 55, 17

25 mod 10 = 5

Index	
0	20
1	
2	
3	
4	
5	25
6	
7	
8	
9	

Hash table

Example: Double hashing

h1(k) = k mod 10

h2(key) = Prime – (key mod Prime)

(h1(key) + i * h2(key)) mod Table size

Elements: 20, 25, 36, 16, 55, 17

36 mod 10 = 6

Index	
0	20
1	
2	
3	
4	
5	25
6	36
7	
8	
9	

Hash table

Example: Double hashing

h1(k) = k mod 10

h2(key) = Prime – (key mod Prime)

(h1(key) + i * h2(key)) mod Table size

Elements: 20, 25, 36, 16, 55, 17

16 mod 10 = 6 --- h1

7- (16 mod 7)

7- (2) =5 --- h2

(6+1*5)mod 10

(11)mod 10 = 1

Index	Hash table
0	20
1	16
2	
3	
4	
5	25
6	36
7	
8	
9	

Example: Double hashing

h1(k) = k mod 10

h2(key) = Prime – (key mod Prime)

(h1(key) + i * h2(key)) mod Table size

Elements: 20, 25, 36, 16, 55, 17

55 mod 10 = 5 --- h1

7- (55 mod 7)

7- (6) =1 --- h2

(5+1*1)mod 10

(6)mod 10 = 6 --------- I

(5+2*1)mod 10 = 7 -------- II

Index	
0	20
1	16
2	
3	
4	
5	25
6	36
7	55
8	
9	

Hash table

Example: Double hashing

h1(k) = k mod 10

h2(key) = Prime – (key mod Prime)

(h1(key) + i * h2(key)) mod Table size

Elements: 20, 25, 36, 16, 55, 17

17 mod 10 = 7 --- h1

7- (17 mod 7)

7- (3) = 4 --- h2

(7+1*4)mod 10

(11)mod 10 = 1 ----------- I

(5+2*4)mod 10 = 3 --------- II

Index	Hash table
0	20
1	16
2	
3	17
4	
5	25
6	36
7	55
8	
9	

Double hashing highlights

- Computational cost is higher
- No primary clustering
- No secondary clustering
- Double hashing can find the next free slot faster than the linear probing approach

Double hashing highlights

- Double hashing is used for uniform distribution of records throughout a hash table.
- Double hashing is useful if an application requires a smaller hash table.

That's all for now…