DATA STRUCTURES AND ITS APPLICATIONS
UE21CS252A

Kusuma K V
Department of Computer Science & Engineering

DATA STRUCTURES AND ITS APPLICATIONS

Hashing: Collision Resolution
Separate Chaining

Kusuma K V
Department of Computer Science & Engineering

Hashing: Separate Chaining

Separate Chaining (Open hashing) handles collision by making every hash table cell point to linked lists of records with identical hash function values.

Hashing: Separate Chaining

Insert 7, 20, 41, 31, 18, 8, 9 into a hash table of size 7.

Use key%tableSize as the hash function and separate chaining (open hashing) to resolve collision.

h(7) = 7 % 7 = 0

h(20) = 20 % 7 = 6

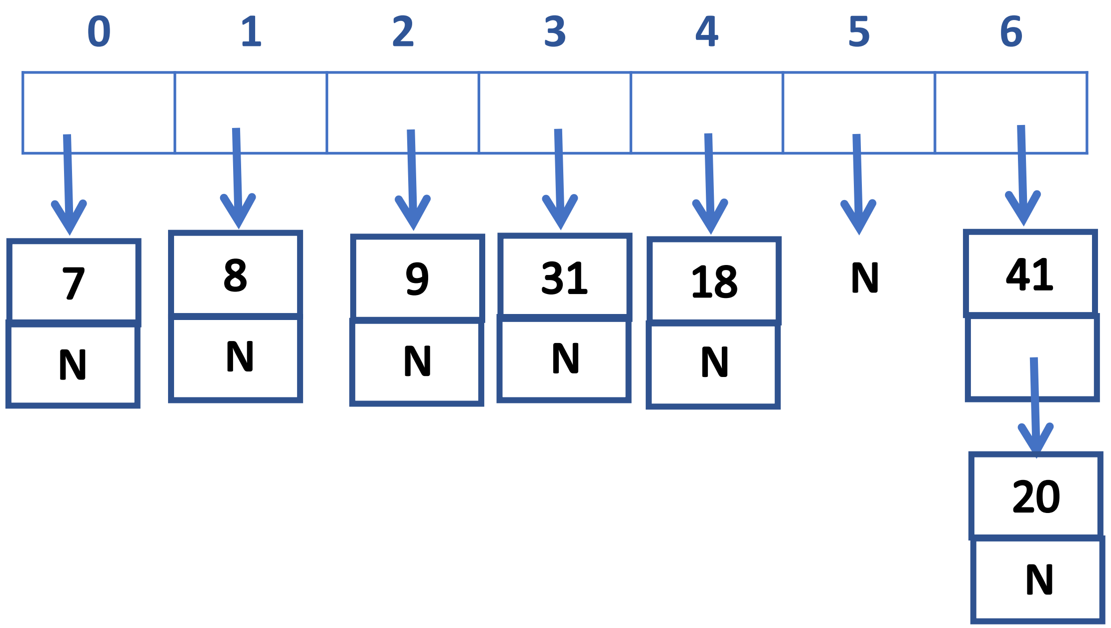
h(41) = 41 % 7 = 6 collision

h(31) = 31 % 7 = 3

h(18) = 18 % 7 = 4

h(8) = 8 % 7 = 1

h(9) = 9 % 7 = 2



Hashing: Separate Chaining

```
typedef struct element
{
        int rNo;
        char name[30];
        struct element *next;
}NODE;
```

Hashing: Separate Chaining

```
void initTable(NODE* ht[SIZE])
{
        for(int i=0;i<SIZE;i++)
                ht[i]=NULL;
}
```

Hashing: Separate Chaining

```
void insert(NODE* ht[SIZE],int rNo,char name[30])
{
        int index=rNo%SIZE;    //hash function

        NODE *newNode=malloc(sizeof(NODE));
        newNode->rNo=rNo;
        strcpy(newNode->name,name);
        newNode->next=ht[index];

        ht[index]=newNode;
}
```

Hashing: Separate Chaining

```
int delete(NODE* ht[SIZE],int rNo)
{
        int index=rNo%SIZE;  //hash function

        NODE *p=ht[index];
        NODE *q=NULL;

        while(p!=NULL && p->rNo!=rNo)
        {
                q=p;
                p=p->next;
        }
        if(p!=NULL)
        {
                if(q==NULL)
                        ht[index]=p->next;
                else
                        q->next=p->next;

                free(p);
                return 1;
        }
        return 0;
}
```

Hashing: Separate Chaining

```
int search(NODE* ht[SIZE],int rNo,char name[30]) {
        int index=rNo%SIZE;    //hash function
        NODE *p=ht[index];
        while(p!=NULL) {
                if(p->rNo==rNo) {
                        strcpy(name,p->name);
                        return 1;
                }
                p=p->next;
        }
        return 0;
}
```

THANK YOU

Kusuma K V

Department of Computer Science
& Engineering

kusumakv@pes.edu