Quadratic probing

#include <stdio.h>

# include<stdlib.h>

#define maxsize 10

int H[maxsize]={NULL};

void insert()

{

int key,hkey,index,i;

printf("enter the key where yo want to inert");

scanf("%d", &key);

hkey=key%maxsize;

for(i=0;i< maxsize;i++)

{

index=hkey%maxsize+(i\*i);

if(H[index] == NULL)

{

H[index]=key;

break;

}

}

}

void display()

{

int i;

printf("\nelements in the hash table are \n");

for(i=0;i<maxsize; i++)

printf("\nat index %d \t value = %d",i,H[i]);

}

int main()

{

int opt,i;

while(1)

{

printf("\nPress 1. Insert\t 2. Display \t3. Search \t4.Exit \n");

scanf("%d",&opt);

switch(opt)

{

case 1:

insert();

break;

case 2:

display();

break;

}

}

return 0;

}

Linear probing

#include <stdio.h>

# include<stdlib.h>

#define maxsize 10

int H[maxsize]={NULL};

void insert()

{

int key,hkey,index,i;

printf("enter the key where yo want to inert");

scanf("%d", &key);

hkey=key%maxsize;

for(i=0;i< maxsize;i++)

{

index=hkey%maxsize+i;

if(H[index] == NULL)

{

H[index]=key;

break;

}

}

}

void display()

{

int i;

printf("\nelements in the hash table are \n");

for(i=0;i<maxsize; i++)

printf("\nat index %d \t value = %d",i,H[i]);

}

int main()

{

int opt,i;

while(1)

{

printf("\nPress 1. Insert\t 2. Display \t3. Search \t4.Exit \n");

scanf("%d",&opt);

switch(opt)

{

case 1:

insert();

break;

case 2:

display();

break;

}

}

return 0;

}