

Raghav Mittal

20M18C0045

B4

25/11/2020

functions of Dictionary using Hashing

void Dictionary :: insert (int key)

{

index = int (key % max)

ptr[index] = (node - type *) malloc (size of
(node - type);

ptr[index] → data = key;

if (root[index] == NULL)

{

root[index] = ptr[index];

root[index] → next = NULL;

temp[index] = ptr[index];

}

else

{

temp[index] = root[index]

while (temp[index] → next != NULL)

temp[index] = temp[index] → next;

temp[index] → next = ptr[index]

}

}

void Dictionary :: search (int key)

{

int flag = 0;

index = int (key % max)

temp[index] = root[index];

while (temp[index] != NULL

if (temp[inden] → data == key)

{

cout << "\n search key is found!!";

flag = 1;

break;

}

else temp[inden] = temp[inden] → next;

}

if (flag == 0)

cout << "\n search key not found";

}

void Dictionary :: delete - del(int key)

{

inden = int (key % max)

temp[inden] = root[inden]

while (temp[inden] → data != key &&
temp[inden] != NULL)

{

ptr[inden] = temp[inden];

temp[inden] = temp[inden] → next

}

ptr[inden] → next = temp[inden] → next
cout << "\n" << temp[inden] → data << "
has been deleted";

temp[inden] → data = -1;

temp[inden] = NULL;

free (temp[inden]);

}