

CS 2103

Assignment in Data Structures # 8

(**Closed Hashing**)

Submitted by:

**Robert Michael Lim, BSCS – 2**

Submitted to:

**Ms. Christine F. Peña**

#define SIZE 20

#define EMPTY '!'

#define DELETED '~'

#define OCCUPIED '@'

typedef struct{

char FN[24], LN[16], MI;

}nameType;

typedef struct{

nameType name;

char course[5];

int yearLevel;

unsigned long idNumber;

}studType;

typedef struct{

studType Student;

char checker; /\*check if deleted or empty or occupied\*/

int next;

}cellType;

typedef struct{

cellType cells[SIZE];

int avail;

}Dictionary;

typedef enum{TRUE,FALSE}Boolean;

int hash(unsigned long x)

{

int sum = 0;

while (x!=0){

sum +=(x%10);

x/=10;

}

return sum % (SIZE/2);

}

Boolean isMember(Dictionary D, studType S)

{

int temp = hash(S.idNumber);

Boolean retval = FALSE;

if(D.cells[temp].checker != EMPTY){

while(temp!=-1 && D.cells[temp].Student.idNumber != S.idNumber){

temp = D.cells[temp].next;

}

if(temp!=-1){

retval = TRUE;

}

}

return retval;

}

void insertDictionary(Dictionary \*D, studType S)

{

int temp, hv;

hv = hash(S.idNumber);

if(isMember((\*D), S)==FALSE){

if(D->cells[hv].checker != OCCUPIED){

D->cells[hv].Student = S;

D->cells[hv].checker = OCCUPIED;

}else{

temp = D->avail;

D->avail = D->cells[temp].next;

D->cells[temp].Student = S;

D->cells[temp].checker = OCCUPIED;

D->cells[temp].next = D->cells[hv].next;

D->cells[hv].next = temp;

}

}

}

void deleteDictionary(Dictionary \*D, studType S)

{

int \*trav, temp, hv;

hv = hash(S.idNumber);

if(isMember((\*D), S) == TRUE){

if(D->cells[hv].Student.idNumber == S.idNumber){

D->cells[hv].checker = DELETED;

}else{

trav = &hv;

while((\*trav)!=-1 && D->cells[(\*trav)].Student.idNumber

!=S.idNumber){

trav = &D->cells[(\*trav)].next;

}

temp = (\*trav);

D->cells[temp].checker =DELETED;

(\*trav) = D->cells[temp].next;

D->cells[temp].next = D->avail;

D->avail = temp;

}

}

}

In this particular variation, it was simple. It is because with the guidance of the Cursor Based Implementation and the review on closed hashing, it was just a matter of combining it. I asked help with Simon on how to do it on most parts but mostly I did it by my own.