חורף 2010 מועד א - פיתרון

1.

פתרון:

א.

פונקציה אחת ">":

a==b => NOT(a<b) AND NOT(b<a)

a>b => NOT(a<b) AND NOT (a==b)

ב.

/\*User functions\*/

typedef BOOL (\*pKeyClone)(pKey pK);

typedef void (\*pKeyDelete)(pKey pK);

typedef BOOL (\*pItemClone)(pItem pI);

typedef void (\*pItemDelete)(pItem pI);

typedef BOOL (\*pLess)(pKey pL, pKey pR);

ג.

/\*Interface functions\*/

Map\* MapCreate(pKeyClone, pKeyDelete, pItemClone, pItemDelete, pLess);

void MapDestroy(Map\* pMap);

BOOL MapInsert(Map\* pMap, pKey pK, pItem pI);

pItem MapFind(Map\* pMap, pKey pK);

BOOL MapRemove(Map\* pMap, pKey pK);

ד.

typedef struct \_pair Pair;

struct \_Map

{

pKeyClone \_pKC;

pKeyDelete \_pKD;

pItemClone \_pIC;

pItemDelete \_pID;

pLess \_pL;

Pair\* \_pHead;

};

ה.

struct \_pair

{

pKey \_pK;

pItem \_pI;

Pair\* \_pNext;

};

BOOL MapInsert(Map\* pMap, pKey pK, pItem pI)

{

Pair\* nP;

if (!pMap)

return FAIL;

if (!(nP = (Pair\*)malloc(sizeof(Pair))))

return FAIL;

if (!(nP->\_pI = (pItem)(pMap->\_pIC(pI))))

return FAIL;

if (!(nP->\_pK = (pKey)(pMap->\_pKC(pK))))

return FAIL;

nP->\_pNext = pMap->\_pHead;

pMap->\_pHead = nP;

}

ו.

BOOL IDClone(pKey pK)

{

int\* pI = (ID\*)malloc(sizeof(ID));

if (!pI)

return FAIL;

\*pI = \*((ID\*)pK);

return SUCCESS;

}

void IDDestroy(pKey pK)

{

if (pK)

free(pK);

}

BOOL PersonClone(pItem pI)

{

Person\* pP;

pP = (Person\*)malloc(sizeof(Person));

if (!nP)

return FAIL;

pP->\_name = strdup(((Person\*)pI)->\_name);

if (!pP->\_name)

return FAIL;

pP->\_addr = strdup(((Person\*)pI)->\_addr);

if (!pP->\_addr)

return SUCCESS;

}

void PersonDelete(pItem pI)

{

if(pI) {

free(((Person\*)pI)->\_addr);

free(((Person\*)pI)->\_name);

free(pI);

}

}

BOOL IDLess(pKey pL, pKey pR)

{

return \*((int\*)pL) < \*((int\*)pR);

}

2

. מבנה נתונים

חלק א:

א.

ניתן לשחזר את העץ תוך שימוש בתכונת הערימה. העץ המתקבל:

1

2 6

3

4 5

ב.

לא ניתן לשחזר כי לא יודעם איך בידיוק לחלק בין התת-עץ השמאלי לימני. יתכן העץ מסעיף א' ויתכן גם:

1

2 3

4

5 6

חלק ב:

ג.

נדרשת ערימה בגלל דרישות סיבוכיות של קבלת/מחיקת האיבר המינימלי. מצד שני, מכיוון שיש חלוקה ברורה לשני פרקי זמן, ובפרק זמן הראשון ההכנסה צריכה להתבצע ב-, ההכנסה בפועל לערימה תתבצע רק בפרק זמן , הבפעלה ראשונה של אחת הפעולות.

בגלל דרישת סיבוכיות מקום, נממש את הערימה המערך. בפרק זמן , פשוט נכניס כל הודעה חדשה למקום הריק הבא במערך (), ללא סידור הערימה. בפרק זמן , לפני ביצוע אחת הפעולות בפעם הראשונה, נבצע פעפוע של האיברים החדשים למקומות הנכונים בערימה, וזאת בסיבוכיות . בכל גישה נוספת לפעולות הנ"ל הסיבוכיות תהיה הסיבוכיות הרגילה של הערימה.

3. C++

הפתרון הוא השורות המודגשות.

א.

**class Page**

**{**

**public:**

**virtual void print() = 0;**

**};**

typedef Page\* PPage;

class Item

{

public:

Item(int num\_pages);

**virtual PPage& operator[](int i) = 0;**

**virtual int getKey() const = 0;**

int getNumPages() const {return num\_pages\_;}

protected:

int num\_pages\_;

PPage\* pages\_;

};

ב.

class Book : public Item

{

public:

// initializes new book.

Book(const char\* name, int id, int num\_pages);

~Book() {delete[] name\_;}

**PPage& operator[](int i);**

**int getKey() const {return id\_;}**

private:

int id\_;

char\* name\_;

};

ד. הבעייה היא ש-2 מחוץ לתחום ההגדרה של הספר הזה. מכיוון שהאופרטור פשוט מחזיר את הרפרנס למצביע המתאים, ניתן לטפל בבעיה זו רק באמצעות חריגות. נממש את האופרטור כך:

**struct PagesException {**

**};**

**Page\*& Book::operator[](int i) {**

**if (i >= num\_pages\_)**

**throw PagesException();**

**return pages\_[i];**

**}**

דוגמא לטיפול בקוד:

**try {**

**b1[2] = &p1;**

**} catch (PagesException e) {**

**cout << "oops" << endl;**

**}**

ה. יש להוסיף אופרטור:

**virtual const PPage& operator[](int i) const = 0;**

או:

**virtual PPage operator[](int i) const = 0;**

כלומר האופרטור צריך להחזיר לפי רפרנס קבוע או לפי ערך (=עותק) את המצביע לעמוד המתאים, במקום להחזיר רפרנס!

4 חלק א פתרון

א.

set len = `echo $1|wc –c`

if ($len > 3 ) echo $1

ב.

foreach file (\*)

if (-f $file) then

if ( $file =~ \*$1\* ) echo $file

endif

end

ג.

set a = `cat $1 | head -5| tail -1 `

echo $a[4]

ד.

grep $1 /etc/passwd | cut –d: -f 5 | cut –d' ' –f2

שאלות בנושא הנדסת תוכנה (9 נק') פתרון

1. כדי שגורם חיצוני לא יוכל ליצור אובייקטים נוספים.
2. בגישה למשאב משותף: למשל תור הדפסות, מערכת הקבצים.
3. כדי שלכולן יהיה ממשק משותף, והסובייקט יוכל להכיר רק ממשק אחד ולעבוד בעזרתו עם כל האובזרוורים.

