# אביב 2007 מועד ב – פתרון

1.

פתרון **(הקטעים ב-bold ימומשו ע"י הסטודנטים**):

קובץ המנשק:

#include <stdlib.h>

#include <malloc.h>

#include <stdio.h>

typedef struct \_Library\* PLibrary;

typedef void\* PItem;

typedef long BarCode;

/\*user-defined functions prototypes:\*/

typedef int (\*ItemHasBarCode)(PItem);

typedef BarCode (\*ItemGetBarCode)(PItem);

typedef void (\*ItemPrint)(PItem);

typedef BarCode (\*GenNewBarCode)();

**/\*interface functions:\*/**

**PLibrary CreateLibrary(ItemHasBarCode, ItemGetBarCode, ItemPrint, GenNewBarCode);**

**void DeleteLibrary(PLibrary);**

**Status LibraryAddItem(PLibrary, PItem, BarCode\*);**

**Status LibraryRemoveItem(PLibrary, PItem);**

**PItem LibraryGetItem(PLibrary, BarCode);**

**void LibraryPrintItems(PLibrary);**

קובץ הממוש:

#include "list.H"

#include "library.H"

#include <stdlib.h>

#include <malloc.h>

#include <stdio.h>

**/\*Library ADT struct definition\*/**

**typedef struct \_Library {**

**PList \_pList;**

**ItemHasBarCode \_fItemHasBarCode;**

**ItemGetBarCode \_fItemGetBarCode;**

**GenNewBarCode \_fGenNewBarCode;**

**ItemPrint \_fPrint;**

**} Library, \*PLibrary;**

/\*Helper struct to be inserted into the PList\*/

**typedef struct \_listItem {**

**PItem \_item;**

**BarCode \_barCode;**

**} ListItem, \*PListItem;**

PKey GetBarCode(PElem pElem)

{

PListItem pItem = (PListItem)pElem;

return (PKey)(pItem->\_barCode);

}

Status CompareBarCodes(PKey pL, PKey pR)

{

long lL, lR;

lL = \*(long\*)pL;

lR = \*(long\*)pR;

return !(lL == lR);

}

/\*interface functions implementation\*/

PLibrary CreateLibrary(ItemHasBarCode pF1, ItemGetBarCode pF2, ItemPrint pF3,

GenNewBarCode pF4)

{

PLibrary pLib = (PLibrary)malloc(sizeof(Library));

if (!pLib) return NULL;

pLib->\_fItemHasBarCode = pF1;

pLib->\_fItemGetBarCode = pF2;

pLib->\_fPrint = pF3;

pLib->\_fGenNewBarCode = pF4;

if (!(pLib->\_pList = CreateList(GetBarCode, CompareBarCodes))) {

free(pLib);

return NULL;

}

return pLib;

}

**Status LibraryAddItem(PLibrary pLib, PItem pItem, BarCode\* pBarCode)**

**{**

**PListItem pNewItem = (PListItem)malloc(sizeof(ListItem));**

**if (!pNewItem)**

**return Failure;**

**pNewItem->\_item = pItem;**

**if (pLib->\_fItemHasBarCode(pItem))**

**pNewItem->\_barCode = pLib->\_fItemGetBarCode(pItem);**

**else**

**pNewItem->\_barCode = pLib->\_fGenNewBarCode();**

**\*pBarCode = pNewItem->\_barCode;**

**return ListAddElem(pLib->\_pList, pNewItem);**

**}**

Status LibraryRemoveItem(PLibrary pLib, PItem pItem)

{

return ListRemoveElem(pLib->\_pList, pItem);

}

PItem LibraryGetItem(PLibrary pLibrary, BarCode barCode)

{

return ListFindElem(pLibrary->\_pList, &barCode);

}

סעיף 4:

typedef struct \_book {

char\* \_name;

char\* \_authour;

char\* \_publisher;

long \_barCode;

int \_hasBarCode; /\*1 - has bar-code; otherwise 0\*/

} Book, \*PBook;

long BookNewBarCode(); /\*Creates new unique BarCode for a Book\*/

void BookPrint(PBook pBook); ; /\*Prints the data stored in Book\*/

**/\*additional functions to be defined by the student\*/**

**int BookHasBarCode(PItem pBook)**

**{**

**return ((PBook)pBook)->\_hasBarCode;**

**}**

**BarCode BookGetBarCode(PItem pBook)**

**{**

**return ((PBook)pBook)->\_barCode;**

**}**

# שאלה 2 - ADT ( 22 נק')

**בשאלה זו אין לכתוב קוד. יש לכתוב הסברים קצרים, ברורים ומדויקים.**

**הסברים קצרים ברורים ומדויקים ניתן לכתוב בעברית, אנגלית, pseudo-code או ע"י תרשימי זרימה.**

**בשאלה זו נעסוק בערימות מינימום, ערימות מקסימום וערימת מינמקס, כפי שנלמדה בתרגיל בית 3. הערימות שלנו ממומשות כעץ בינארי כמעט מלא המסודר לפי חוק הערימה (החוק המתאים לכל אחת מסוגי הערימות). העץ הינו מוקצה דינאמית וכל צומת מכילה (פרט לנתונים) מצביע לאב ולשני הבנים; מצביעים אלה יקראו father, left,right, בהתאמה.**

**תזכורות:**

**1) ערימת מינמקס היא מבנה נתונים המבוסס על מבנה הערימה המספק את הפעולות הבאות, n הינו מספר האיברים בערימה:**

* **אתחול כערימת מינמקס ריקה, סיבוכיות O(1)**
* **הוספת איבר, סיבוכיות O(log n)**
* **מציאת האיבר המקסימלי, סיבוכיות O(1)**
* **מציאת האיבר המינימלי, סיבוכיות O(1)**
* **הוצאת האיבר המקסימלי, סיבוכיות O(log n)**
* **הוצאת האיבר המינימלי, סיבוכיות O(log n)**

**2) במבחן מועד א' ראינו כי בהינתן ערימת המקסימום הממומשת כעץ מוקצה דינאמית כפי שאצלנו ומצביע לצומת בערימה, ניתן להוציא צומת זאת מהערימה בסיבוכיות O(log n). אלגוריתם זה אינו צורך זכרון נוסף לערימה וניתן להתאמה גם לערימת מינימום. בשאלה זאת אין צורך לדעת כיצד עובד אלגוריתם זה, אך ניתו להניח את קיומו (הניחו כי מסופקת לכן פונק' מתאימה) ולהשתמש בו בשם "האלגוריתם שמוציא צומת מערימה".**

**א. (8 נק')**

**בחלק א' של ת"ב 3 הראיתם כי לא ניתן להחליף ערימת מינמקס בערימת מינימום וערימת מקסימום תוך כדי שמירת הסיבוכיות של הפעולות. באיזו פעולות נוצרת בעיה של סיבוכיות? (1 נק')**

***הבעיה הינה במימוש שתי הפונקציות האחרונות כלומר הוצאת המינימום והמקסימום.***

**כיצד תממשו פעולות אלו (בדרך היעילה ביותר) ומהי סיבוכיות הזמן והמקום הנוסף (בנוסף לשתי ערימות) בשני המקרים הבאים:**

1. **ערימות המינימום והמקסימום מסופקות לכן כ"קופסא שחורה" (נניח ADT או class ב-C++) וניתן לבצע אך ורק את הפעולות הסטנדרטיות של ערימות בסיבוכיות כפי שנלמדה בהרצאה. (4 נק')**

***במקרה זה ע"מ להוציא למשל את איבר המינימלי מערימת המקסימום יש לנקוט בצעדים הבאים:***

* 1. ***להוציאו מערימת המינימום – O(log n)***
  2. ***להוציא את כל אברי ערימת המקסימום ולאכסנם בצד (O(n\*log n) זמן + O(n) מקום)***
  3. ***לבנות מחדש את ערימת המקסימום ע"י הכנסה איבר איבר (O(n\*log n))***

***סה"כ (O(n\*log n) זמן + O(n) מקום נוסף.***

**ערימות המינימום והמקסימום הינן "קופסא לבנה" (נניח שאתם מימשתם אותם) וניתן לגשת למבנה העץ הפנימי. (3 נק').**

***במקרה זה ע"מ להוציא למשל את האיבר המינימלי לנקוט בצעדים הבאים:***

1. ***להוציאו מערימת המינימום – O(log n)***
2. ***לבצע סיור בעץ של ערימת המקסימום ולמצאו שם O(n)***
3. ***להוציא לערימת המקסימום ע"י האלגוריתם שמוציא צומת מערימה – O(log n)***

***סה"כ O(n) זמן ו O(1) מקום.***

**מעתה והלאה הניחו כי כל המבנים נכתבים על ידיכם ויש לכם גישה למבנה הפנימי.**

**ב. (12 נק')**

1. **הציעו שינוי קל שיאפשר להחליף את ערימת המינמקס בערימת מינימום וערימת מקסימום. כלומר, הסבירו איזה שינוי/תוספת יש לבצע לערימות על מנת שניתן יהיה לממש את כל הפונקציות באותה סיבוכיות כמו ערימת מינמקס מהי סיבוכיות הזיכרון שנוספת לשתי הערימות בעקבות השינויים שלכם? מהי ולכן סיבוכיות המקום של המבנה הכולל (כולל שתי הערימות והשינויים שעשיתם)? השוו זאת לסיבוכיות הזיכרון של ערימת מינמקס יחידה. (5 נק')**

***ע"מ לאפשר את 2 הפעולות יש להוסיף בכל צומת של שתי הערימות מצביע נוסף שיצביע על הצומת המכיל את אותו איבר בערימה השניה, סיבוכיות המקום שנוספת היא O(n) ולכן הכוללת גם היא O(n) בדיוק כמו ערימת מינמקס יחידה.***

1. **כיצד תממשו הכנסת איבר חדש למבנה? (אין צורך להסביר את אלגוריתם הכנסת איבר לערימה). (3 נק')**
   1. ***נכניס אותו לערימת המקסימום ללא איתחול המצביע לערימת המינימום ונשמור מצביע לצומת שלו בערימת המקסימום. O(log n)***
   2. ***נכניס אותו לערימת המינימום ונאתחל את המצביע לערימת המקסימום לפי הערך ששמרנו בשלב קודם. נשמור מצביע לצומת שלו בערימת המינימום. O(log n)***
   3. ***ע"י המצביע לצומת בערימת המקסימום ניגש ישירות לצומת זה ונאתחל את המצביא לצומת המתאים בערימת המינימום. O(1)***
2. **כיצד תממשו את הוצאת האיבר המינימלי? (4 נק')**
   1. ***נוציא אותו מערימת המינימום ושמור את המצביע למיקומו בערימת המקסימום O(log n).***
   2. ***ע"י המצביע לצומת בערימת המקסימום נשתמש באלגוריתם המוציא צומת מערימה ע"מ להוציאו מערימת המקסימום.***

**פתרון שאלה 3**

חלק א':

1. מסתירה כי החתימה שונה.
2. מסתירה כי החתימה שונה.
3. מסתירה כי המתודה  ב- אינה וירטואלית.
4. להלן:

B::f

B::h

B::h

C::h

A::g

C::g

C::g

חלק ב':

class Complex;

class Complex Ptr {

public:

ComplexPtr(Complex\* p) : p\_(p) { **++p\_->refCount\_;** }

~ComplexPtr() { **if (--p\_->refCount\_ == 0) delete p\_;** }

ComplexPtr(const ComplexPtr& p) : p\_(p.p\_) { **++p\_->refCount\_;** }

Complex& operator\* () { **return \*p\_;** }

ComplexPtr& operator= (const ComplexPtr& p)

{

**Complex\* const old = p\_;**

**p\_ = p.p\_;**

**++p\_->refCount\_;**

**if (--old->refCount\_ == 0) delete old;**

**return \*this;**

}

private:

Complex\* p\_;

};

1. ניתן להגדיר את כל סוגי  כפרטיים, ולהגדיר פונקציית יצירה ציבורית, אשר משתמשת בהקצאה דינמית ומחזירה "מצביע" למטריצה (כלומר אוביקט מסוג ) .

**פתרון שאלה 4**

א.

cat books |sort|uniq|cut -d':' -f1,3

ב.

#!/bin/csh

set lines = 0

if ($argv[1] == "-all") then

foreach file ( \* )

if (-f $file) then

set tot = `wc -l $file`

@ lines = $lines + $tot[1]

endif

end

else if ($argv[1] == "-these\_files") then

shift

foreach file ( $\*)

if (-f $file) then

set tot = `wc -l $file`

@ lines = $lines + $tot[1]

endif

end

endif

echo $lines