

Labc

תרגיל 3 - מבני נתונים וקצת אלגוריתמיקה

הגשה: יום ראשון ה- 21.11.04

1. (15%) היפטרות מהתייחסות למקרי קצה

התבוננו בקוד הנתון בקובץ q1.c. כפי שניתן לראות, קוד זה יוצר רשימה משורשרת ממוינת בת חמישים אלף מספרים אקראיים ולאחר מכן מדפיס את שבע עשרה האברים הראשונים בה. כמו כן מדפיסה התוכנית את מספר השניות שלקח לה לבנות את הרשימה הממוינת. עליכם להחליף את מימוש הפונקציה sortedInsert בפונקציה קצרה יותר. הפונקציה הקיימת מפרידה בין ארבעה מיקרים שונים, אחד מהם המקרה הטיפוסי ושלושת האחרים - מקרי קצה. הפונקציה שעליכם לכתוב אמורה להתייחס רק למקרה אחד (הטיפוסי). במקום 21 שורות הקוד הקיימות, אמורות להספיק 5 או 6. חוץ מהחלפת מימוש הפונקציה אין לבצע שום שינוי נוסף בקוד.

- האם יש שיפור גם ביעילות הקוד? אם כן, מדוע? אם לא, מדוע?

2. (20%) מערך גמיש

בשאלה זו תתבקשו לכתוב מבנה נתונים בשם Vector המתנהג כמו מערך גמיש. כאשר מבנה הנתונים נוצר הוא יחזיק מערך ריק בגודל שני בתים. מבנה הנתונים יאפשר הוספת תווים חדשים לסופו. במידה והמערך המוחזק מתמלא, יש לדאוג להגדלתו ע"י שימוש בפונקציה הספרייה הסטנדרטית - realloc (ניתן לקרוא ב - man 3 realloc, ב - K&R או ברשת). הקובץ q2.c מכיל קוד המשתמש במבנה הנתונים המבוקש. הקובץ q2.out מגדיר את פורמט הפלט המבוקש. שימו לב שהדפסת הווקטור מתייחסת גם לתאים הריקים (קונספטואלית) במבנה.

3. (65%) כיווץ קבצים

בשאלה זו אתם מתבקשים לכתוב אפליקציה המסוגלת לכווץ קבצים. התוכנית תשתמש באלגוריתם של Hoffman למציאת קידוד חסר רישא. אם קובץ הריצה ניקרא q3, בכדי לכווץ קובץ בשם file.stam יש לכתוב:

```
$ q3 c file.stam
```

(c בשביל 'compress')

פקודה זו תגרום ליצירת קובץ מכווץ בשם file.stam.compressed. בכדי לשחזר את הקובץ המקורי מהקובץ המכווץ יש לכתוב:

```
$ q3 x file.stam.compressed
```

(x בשביל 'extract')

פקודה זו תגרום ליצירת קובץ בשם file.stam.decompressed . קובץ זה אמור להיות זהה לקובץ המקורי. הפקודה:

```
$ diff file.stam file.stam.decompressed
```

אמורה לא להדפיס דבר.

שימו לב שהקובץ file.stam איננו בהכרח קובץ טקסט, הוא יכל להיות קובץ הרצה, תמונה או כל דבר אחר.

ניתן יהיה גם לבקש מהתוכנית להדפיס אינפורמציה נוספת לפלט הסטנדרטי. הפקודה:

```
$ q3 cv file.stam
```

תכנון את הקובץ כמו פקודת הכיוון הקודמת ובנוסף תדפיס על המסך את השכיחויות של התווים השונים, את עץ הקידוד שייצרה ואת הקוד שניתן לכל תו. הפקודה:

```
$ q3 xv file.stam.compressed
```

תשחזר את הקובץ כמו פקודת השחזור הקודמת ותדפיס על המסך את עץ הקידוד. קובץ ההרצה q3 שבתיקיית התרגיל מדגים את ההתנהגות המבוקשת. אין הכרח להדפיס הדפסות זהות לאלה של q3 אך חשוב שניתן יהיה לראות בהדפסות שלכם את עץ הקידוד ואת הקוד המתאים לכל תו.

פרוט והדרכה:

השלב הראשון בכיוון הוא מעבר על הקובץ וספירת המופעים של כל תו. לאחר שאינפורמציה זו הושגה אנו ממליצים להשתמש במבנה היכול לשמש כקודקוד ברשימה וכקודקוד בעץ:

```
typedef struct node {  
    unsigned char m_letter;  
    int m_count;  
    struct node *m_next,*m_ls,*m_rs;  
} Node;
```

כפי שניתן לראות, בקודקוד שמור גם התו (m_letter) וגם מספר המופעים שלו (m_count). אם מתחילים מרשימה ממוינת לפי m_count. אפשר להוציא בכל שלב את שני הקודקודים בעלי המופעים הקטנים ביותר, ליצור קודקוד חדש שהם יהיו בניו, מספר המופעים שלו יהיה סכומם ואז להוסיף את הקודקוד החדש למקומו הממוין ברשימה. בסופו של התהליך תכיל הרשימה קודקוד בודד - שורש העץ.

מבחינת יעילות עדיף כמובן להשתמש במבנה נתונים התומך בשליפה מהירה של האבר הקטן ביותר ולא ברשימה משורשרת פשוטה. מבנה של ערמה הוא בחירה טבעית. מימוש ערמה על מנת לייעל את התהליך ייחשב בונס (עד 7 נק').

השלב הבא הוא לקודד את העץ לקובץ המכווץ. הדבר הכרחי מכיוון שללא מפתח לקידוד, לא ניתן יהיה לפענח את הקובץ. קידוד עץ ניתן לעשות באופן דומה להדפסת תכולת עץ למסך. לאחר כתיבת העץ לקובץ המטרה יש לכתוב את קידוד ה-hoffman של כל תו מקובץ המקור. העץ הוא כמובן לא מבנה נתונים נח לשם כך וכדאי לבנות מערך מיפוי מתאים.

השלב הראשון בשחזור קובץ מכווץ, הוא קריאה ושחזור של עץ הקידוד המצוי בראשיתו. שיחזור של העץ אמור להיות דומה לקידוד שלו עם תוספת מסוימת. לאחר שיחזור העץ אפשר לקרוא את שאר הקובץ ולפענח אותו בעזרת העץ.

גם בקידוד וגם בפענוח העבודה היא ברמת ביטים בודדים. מכיוון שלא ניתן לכתוב ולקרא מקבצים פחות מ-byte שלם, התוכנית תבצע buffering ברמה של byte בודד. בכדי לשלוט בביטים של byte ניתן להשתמש באופרטורים לעבודה עם ביטים. הקובץ bits.c שבתיקיית התרגיל מדגים שימוש באופרטורים אלה (כדאי להריץ). הקובץ sample.txt שבתיקיית התרגיל יכול לשמש כקובץ בדיקה לכיווץ.

בונוס (עד 12 נק'): ממשו את אלגוריתם הדחיסה של Lempel-Ziv. על הקובץ להיקרא q4.c ועל התוכנית לענות לאותו ממשק הפעלה של q3.

הגשה:

קובץ tgz הכולל:

q1.c המכיל את התוכנית הנתונה לאחר החלפת הפונקציה המבוקשת.

q2.c ה-main הנתון בתוספת הקוד הגורם לפלט הדרוש.

q3.c המתקמפלת לתוכנית שקולה ל-q3 המסופקת.

q4.c (אופציונאלי) בונוס

readme לפי ההנחיות הרגילות. לא לשכוח לציין אם מימשתם בונוס.

(נא להקפיד על שמות קבצים אלה)

תדפיסים לתא בתוספת התשובה לשאלת היעילות שבשאלה הראשונה.

בהצלחה !