

מבני נתונים, סמסטר ב' תש"פ
המכללה האקדמית של תל-אביב-יפו

תרגיל תכנות 2

נושא התרגיל: בעיית הדפסת ערכים הקטנים מערך נתון

תאריך הגשה: יום שני 1.6.20 בחצות.

הנחיות כלליות

1. התרגיל הינו תרגיל חובה.
2. התרגיל ניתן להגשה בבודדים או בזוגות, אך לא בקבוצות גדולות יותר.
3. גם תלמידים החוזרים על הקורס מחויבים בהגשת התרגיל, ולא משנות הנסיבות.
4. ציון "נכשל" בתרגיל, או הגשה מאוחרת שלו, ללא אישור מתאים שניתן על ידי המרצה, יגררו ציון "נכשל" סופי בקורס.
5. איחור בהגשה יאושר רק במקרה של מילואים, מחלה ממושכת או לידה, וגם זאת רק בתנאי שהפנייה למרצה בנושא נעשתה לפני מועד ההגשה המקורי של התרגיל.

מטרת התרגיל

עליכם לכתוב תכנית בשפת C++ ובה בלבד, שבה תיפתר בעיית הדפסת ערכים קטנים מערך נתון המתוארת להלן בכמה דרכים, לפי כמה אלגוריתמים שלמדנו בקורס, ותבצע השוואה בין יעילות הפתרונות השונים.

בעיית הדפסת ערכים הקטנים מערך נתון

בהינתן n איברים כאשר כל איבר מכיל מפתח ונתונים, ומפתח נוסף K , יש להדפיס את כל הנתונים עם מפתחות הקטנים מ- K ממוינים מהקטן לגדול.
הנחה: כל המפתחות שונים זה מזה.

השלבים לביצוע התרגיל

- א. קראו תחילה היטב את ההנחיות של התרגיל והבינו מה נדרש בדיוק.
- ב. תכננו את ה-design של התוכנית שלכם: בחרו אילו מחלקות תממשו, החליטו על data members מתאימים ועל methods רלוונטיים לכל מחלקה.
- ג. כתבו מימוש מלא לכל המחלקות שייעשה בהן שימוש במסגרת התוכנית.
- ד. אסור השימוש במחלקות map, set של STL אך מותר להשתמש ביתר המחלקות של STL (בפרט string). במידה ולא תשתמשו ב-string הקפידו להקצות מערך בגודל 128 תווים עבור השם.

תיאור מפורט של מטרת התרגיל:

מטרת התרגיל היא לממש מספר פתרונות לבעיית הדפסת ערכים קטנים מערך נתון, לחקור ולנתח את ההבדלים בין הפתרונות השונים ביחס ל n, r כאשר r הוא מספר הנתונים המודפסים. הפתרונות שעליכם לממש הם:

1. `int NaivePrint(Student [], int n, int k)`

ממשו אלגוריתם שעובר על המערך איבר איבר, ומעתיק ממנו את כל הנתונים בעלי מפתח קטן מ- k , לרשימה משורשרת כאשר ההכנסה מתבצעת באופן ממין (הרשימה ממוינת בכל רגע). לאחר מכן תודפס הרשימה למסך.

2. `int BSTPrint(int [], int n, int k)`

ממשו אלגוריתם שבונה עץ חיפוש בינארי (רגיל, לא מאוזן) מהאיברים, כאשר מכניסים את כל האיברים אחד אחד לעץ, ואז מדפיס את האיברים inorder עד שמגיע לערך גדול או שווה ל- k .

3. `int PrintBySort(int [], int n, int k)`

ממשו אלגוריתם שממין את המערך במיון quick-sort (אותו עליכם לממש), ואז מדפיס את האיברים עד שהוא נתקל בערך גדול או שווה ל- k .

דרישות

- עליכם לממש כל אחת משלוש הפונקציות המתוארות לעיל, כאשר כל פונקציה מקבלת מערך בגודל n של איברים שונים זה מזה, את גודל n ואת הערך K .
- בנוסף לפתרון הבעיה, הפונקציות צריכות לספור את כמות ההשוואות שמבצע כל אלגוריתם על המערך (השוואה נספרת רק בין 2 איברי קלט. השוואות אחרות, כמו השוואה בין אינדקסים, לא נחשבות).
- מספר ההשוואות שביצעה הפונקציה יהיה הערך המוחזר ממנה.

התכנית הראשית :

פונקציית main של התכנית:

1. תקרא מ-stdin (מהמקלדת, לא מקובץ) מספר שלם, n , שמסמן את מספר האיברים, ולאחר מכן תקרא n צמדים של מספר ואחריו מחרוזת, כל אחד בשורה. המספר השלם בכל זוג מסמן מספר זהות של שחקן, והמחרוזת את שמו. יש לבדוק שכל שם אכן מורכב מ-2 מילים מופרדות ברווח (שם פרטי, רווח, שם משפחה).
לאחר מכן תקלוט מספר שלם, k .
למשל, הקלט הבא:

```
6
132 LeBron James
118 Stephen Curry
313 Kevin Durant
514 Michael Jordan
323 Dennis Rodman
606 Scottie Pippen
318
```

הוא קלט של שישה שחקנים, כאשר מפתח המיון הוא מספר הזהות שלהם, והשם הוא נתוני הלוויין שלהם. יש להדפיס את כל הנתונים שמספר הזהות שלהם קטן מ-318 באופן ממין ולכן יש להדפיס למסך:

```
118 Stephen Curry
132 LeBron James
313 Kevin Durant
```

2. תריץ כל אחת משלוש הפונקציות המתוארות למעלה על סדרת הנתונים ותקבל מכל פונקציה את מספר ההשוואות שבוצעו נוסף על ההדפסות שקורות בה. שימו לב: הנתונים הממוינים יודפסו 3 פעמים.
3. תדפיס למסך את מספר ההשוואות בפורמט הבא:

NaivePrint: 130 comparisons

BSTPrint: 200 comparisons

PrintBySort: 500 comparisons

(המספרים של ההשוואות שרשומים בדוגמה כמובן מפוברקים).

מבני הנתונים

לצורך ביצוע התרגיל, הנכם נדרשים לממש את המחלקה עץ חיפוש בינארי ללא שימוש ב-STL.

עליכם לממש את הפעולות הבסיסיות שמוגדרות על עץ חיפוש בינארי, גם אם הן לא נדרשות לצורך התכנית: IsEmpty, MakeEmpty Insert, Delete, Find.

כמו כן עליכם לממש כמובן כל מחלקה אחרת שלה תזדקקו במהלך התכנית.

סיבוכיות וביצוע בדיקות ותייעוד:

עליכם לייצר טבלה שתכיל את ניסיונות הרצה של התכנית שלכם על 10 קלטים שונים.

כל אחד מהקלטים יכיל את אותם $n=100$ נתונים. הקלט ה- i יכיל k שהוא $10i$ בגודלו (כלומר יגרום להדפסה של $10i$ נתונים).

ואז סיכום התוצאות באופן הבא:

- בטבלה יהיו 3 עמודות: עמודה לכל אלגוריתם.
- בטבלה יהיו 10 שורות שיתאימו ל-10 הקלטים שתוארו, כאשר בכל שורה יירשמו מספר ההשוואות של כל אחד מהאלגוריתם על הקלט שבאותה שורה. להלן מבנה הטבלה:

	<u>NaivePrint</u>	<u>BSTPrint</u>	<u>PrintBySort</u>
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

את הטבלה הגישו בקובץ PDF.

בדיקת שגיאות:

הקפידו לבדוק שגיאות אפשריות בקלט. כך למשל שגיאות יכולות להכיל קלט שמכיל שני שחקנים עם אותו מספר זהות, וכך הלאה. במקרה של שגיאה כתבו הודעת שגיאה למסך invalid input וצאו מהתכנית באמצעות הפונקציה `exit(1)`.

(יש לבצע `#include <stdlib.h>` על מנת להשתמש בה).

הנחיות הגשה

יש להגיש במערכת mama במקום המיועד להגשה את הקבצים הבאים:

1. קובץ readme שיכיל את כל פרטי ההגשה הבאים:

כותרת – תרגיל תכנות מס' 2 במבני נתונים תש"ף סמסטר ב'.

שורה מתחת - שמות המגישים ומספרי זהות שלהם.

שימו לב: קובץ טקסט פשוט – לא word.

2. כל קבצי הקוד בסיומות cpp. ו- h.

3. קובץ PDF עם הטבלה שמכילה את תוצאות ההרצות.

שימו לב! הגשה שאינה בפורמט הנדרש תידחה אוטומטית.

הערות הכרחיות נוספות

- הקפידו על תיעוד, שמות בעלי משמעות למשתנים, מודולאריות וכל מה שנדרש מתכנות נכון. בתיעוד בראש התוכנית כתבו גם הוראות הפעלה מדויקות וברורות.
- הקפידו על חלוקה נכונה לקבצים (קבצי CPP וקבצי H לכל מחלקה).
- תכנתו Object Oriented והימנעו מאלמנטים מיותרים של קוד פרוצדוראלי.
- הקפידו לשחרר את כל הזיכרון אשר הקציתם דינמית לאחר שהשתמשתם בו.
- במקרה של תקלה בריצת התוכנית (מסיבה כלשהי), עליה לדווח זאת למשתמש טרם סיימה לרוץ.
- בדקו את תכניותיכם על קלטים רבים ככל האפשר - כולל קלטים חוקיים ולא חוקיים, וזאת בנוסף לטסטים שהוכנו עבורכם במאמא.
- **תנאי הכרחי (אך כמובן לא מספיק) לקבלת ציון עובר על התרגיל, הוא שהקוד יעבור קומפילציה. ציונו של תרגיל אשר אינו עובר קומפילציה יהיה 0.**

בהצלחה!