

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

תרגיל תכנות 2
נושא התרגיל: בעיית הבחירה Selection

תאריך הגשה: יום חמישי ה- 16.1.20 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. בחירת השפה אינה נתונה לשיקול דעת הסטודנטים), שבה תיפתר בעיית הבחירה selection בכמה דרכים, לפי כמה אלגוריתמים שלמדנו בקורס, ותבצע השוואה בין יעילות הפתרונות השונים.

בעיית הבחירה Selection:

בהינתן n איברים שונים זה מזה, כאשר כל איבר מכיל מפתח ונתונים, יש למצוא את האיבר עם המפתח ה k בגודלו (כלומר, האיבר שיש בדיוק k-1 איברים עם מפתחות הקטנים ממנו).

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 שיעשה בהן שימוש במסגרת התוכנית.
- ד. אסור השימוש במחלקות map, set של STL, אך מותר להשתמש ביתר המחלקות של STL.

תיאור מפורט של מטרת התרגיל:

מטרת הפרויקט היא לממש מספר פתרונות לבעיית הבחירה Selection, לחקור ולנתח את ההבדלים בין הפתרונות השונים ביחס ל n, k . הפתרונות שעליכם לממש הם:

1. const Person & RandSelection(Person [], int k, int &NumComp)
ממשו את אלגוריתם Selection כפי שמופיע במקראה, אבל כאשר בחירת ה-pivot בכל שלב היא **אקראית** (בעזרת הפונקציה rand).
2. const Person & selectHeap(Person [], int k, int &NumComp)
יצירת ערימת מינימום בינארית מ- n הפריטים שממומשת במערך כפי שנלמד בכיתה (כאשר הסדר הוא לפי המפתח כמובן), וביצוע k פעמים deleteMin (שימו לב שכאן מורידים את האיברים מהערימה כדי למצוא את האיבר ה- k . אין צורך להחזיר אותם לערימה).
3. const Person & BST(Person [], int k, int &NumComp)
יצירת עץ חיפוש בינארי (רגיל, לא מאוזן) מהאיברים, כאשר מכניסים את האיברים אחד אחד לעץ, ואז מציאת האיבר ה- k בגודלו בעץ (ללא הורדתו מהעץ!).

דרישות

1. עליכם לממש כל אחת משלוש הפונקציות המתוארות לעיל, כאשר כל פונקציה מקבלת מערך בגודל n של איברים שונים זה מזה מסוג Person ואינדקס k .
2. 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).
3. הערך המוחזר מכל פונקציה יהיה אובייקט מסוג Person הכולל את המפתח ויתר הנתונים של האדם, ובפרמטר פלט NumComp יאוחסן מספר ההשוואות שביצעה הפונקציה.

התכנית הראשית :

פונקציית main של התכנית:

1. תקרא מ-stdin מספר שלם, n , שמסמן את מספר האיברים, ולאחר מכן יקראו n צמדים של מספר ואחריו מחרוזת, כל צמד בשורה נפרדת. המספר השלם בכל זוג מסמן מספר זהות של אדם, והמחרוזת את שמו. לאחר מכן התכנית תקלוט מספר שלם, k , שהוא בין 1 ל- n . למשל, הקלט הבא:

```
3
132 LeBron James
118 Stephen Curry
313 Kevin Durant
2
```

הוא קלט של שלושה אנשים, כאשר מפתח המיון הוא מספר הזהות שלהם, והשם הוא נתוני הלוויין שלהם. בדוגמה זו, הראשון במיון הוא 118, השני 132 והשלישי 313. המשתמש מעוניין בשני במיון, כלומר $k = 2$.

2. תריץ כל אחת משלוש הפונקציות המתוארות למעלה על סדרת הנתונים ותקבל מכל פונקציה את מספר ההשוואות שבוצעו ואת נתוני האיבר k בגודלו.

3. תדפיס למסך את המפתח והנתונים של האדם שהוא האיבר k , ואת מספר ההשוואות שביצעה כל אחת משלוש הפונקציות, בפורמט הבא:

132 LeBron James

RandSelection: 130 comparisons

selectHeap: 200 comparisons

BST: 500 comparisons

(המספרים של ההשוואות שרשומים בדוגמה כמובן מפוברקים).

מבני הנתונים

לצורך ביצוע התרגיל, הנכם נדרשים לממש את המחלקות הבאות ללא שימוש ב-STL:

1. עץ חיפוש בינארי.
2. ערימת מינימום.

עליכם לממש לכל אחד ממבני הנתונים האלה את הפעולות הבסיסיות שמוגדרות עליו, גם אם הן לא נדרשות לצורך התכנית.

למשל, לעץ חיפוש בינארי ממשו את Insert, Delete, Find ולערימת מינימום את Insert, DeleteMin, Min BuildHeap ו-1 של פלויד, וכמובן גם את הפונקציות IsEmpty, MakeEmpty.

שימו לב שגם בחלק מהפעולות האלה שמוגדרות על העץ ועל הערימה יש השוואות בין האיברים, למשל ב- Insert או ב- BuildHeap, ולכן חשוב לספור גם אותן.

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

סיבוכיות וביצוע בדיקות ותיעוד:

יש לתעד בקוד את סיבוכיות זמן הריצה במקרה הגרוע ובממוצע (האסימפטוטית, במונחי Θ הדוקים) של כל אחת משלוש הפונקציות המרכזיות, כתלות ב- n ו- k . עליכם להשיג סיבוכיות זמן ריצה (במקרה הגרוע ביותר) נמוכה ככל הניתן עבור כל אחת מהפונקציות.

בנוסף עליכם לייצר טבלה שתכיל את ניסיונות הרצה של התכנית שלכם על 10 קלטים שונים, כל אחד באורך 100, ואז סיכום התוצאות באופן הבא:

- בטבלה יהיו 6 עמודות: עמודה אחת לאלגוריתם selectHeap, עמודה אחת לאלגוריתם BST ו- 4 עמודות לאלגוריתם RandSelection.
- בטבלה יהיו 11 שורות: 10 השורות הראשונות יתאימו ל- 10 קלטים שונים, כאשר בכל שורה יירשמו מספר ההשוואות של כל אחד מהאלגוריתמים על הקלט שבאותה שורה. בשורה אחרונה עליכם לסכם את ממוצע ההשוואות של כל אלגוריתם על 10 הקלטים, כאשר במקרה של RandSelection עליכם להריץ 4 פעמים את הפונקציה על כל קלט, מכיוון שהשיטה היא הסתברותית וזאת כי ה-pivot נבחר באקראי, ואז לחשב ממוצע אחד ל- 40 התוצאות של RandSelection.

הערה: שימו לב שכאשר תריצו את התכנית 4 פעמים על קלט מסוים לטובת קבלת התוצאות ל- RandSelection, תקבלו שוב ושוב גם את התוצאות של BST ושל selectHeap, אבל אלה לא ישתנו כמובן מכיוון ששני האלגוריתמים האלה לא הסתברותיים, ולכן מספיק לתעד את התוצאה שלהם על כל קלט רק פעם אחת.

להלן מבנה הטבלה:

	selectHeap	BST	RandSelection1	RandSelection2	RandSelection3	RandSelection4
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
Average						

10 הקלטים שעליכם לבדוק ולתעד את תוצאות הריצה שלהם בטבלה נמצאים במאמא, והם כוללים: קלט שממוין מראש בסדר עולה, קלט שממוין מראש בסדר יורד, ועוד 8 קלטים אחרים שנמצאים כאמור במאמא.

את הטבלה הגישו בקובץ PDF.

בדיקת שגיאות:

הקפידו לבדוק שגיאות אפשריות בקלט. כך למשל, שגיאות יכולות להכיל קלט שמכיל שני אנשים 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 א'.

שורה מתחת - שמות המגשים, מספרי ת.ז. שלהם ומספר הקבוצה של כל אחד מהם (מותר להגיש עם בן זוג מקבוצה אחרת).

שימו לב: קובץ טקסט פשוט – לא word.

2. כל קבצי הקוד בסיומות `.cpp`. 1- `.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 שהשתמשם 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.**

צוות מבני נתונים, סמסטר א' תש"ף