

מבני נתונים 89-120

תרגיל 10

גלעד אשרוב צבי קופולביץ'

31 במאי 2010

ההגשה ביחידים. כל סטודנט נדרש לחשוב, לפתור ולכתוב את התרגיל בעצמו. פותר להתייעץ עם סטודנטים אחרים - רק אחרי שניסית בכל כוחך לשבת על התרגיל לבד. בכל אופן, חל איסור פוחלט להסתכל על תרגיל כתוב של אחר, וחובה על כל סטודנט לכתוב את התרגיל לבדו.

תאריך הגשה: לשיעור התרגול הבא (תרגול 12), בקבוצת התרגול - בין התאריכים 06.06 ל - 09.06.

בכל השאלות הדורשות להציג אלגוריתם - יש להסביר במילים כיצד האלגוריתם עובד, ומדוע הוא עובד. הנימוקים צריכים להיות משכנעים.

שאלה 1. הכניסו את הקודקודים הבאים לעץ 2-3 ריק: 10, 40, 35, 25, 60, 30, 80, 50, 27, 38 (משמאל לימין). ציירו את העץ המתקבל לאחר כל הכנסה.

שאלה 2. נתון עץ B-tree מדרגה m המכיל k איברים. הוכח או הפרך: אם נכניס איבר נוסף לעץ, ובשלב הבא נבטל אותו (נוציא אותו), העץ המתקבל לאחר ביצוע שתי הפעולות יהיה זהה לעץ המקורי שאיתו התחלנו.

שאלה 3. בהינתן k מספרים כלשהם, נגדיר את החציון להיות האיבר ה- $k/2$ בגודלו.

(א) בהינתן שני מערכים ממויינים מגודל n כל אחד, מצא את החציון של $2n$ איברי שני המערכים בזמן $O(\log n)$. לשם פשטות, ניתן להניח ש- n חזקה שלמה של 2.

(ב) בהינתן שני מערכים, האחד בגודל n והשני בגודל m , מצא את החציון של $n + m$ איברי שני המערכים בזמן $O(\log(n + m))$. ניתן להניח ש- n ו- m חזקות שלמות של 2.

שאלה 4.

(א) בעץ חיפוש בינארי דיברנו על הפעולות *insert*, *delete* ו-*search*. נרצה בנוסף לפעולות הרגילות לתמוך בפעולה הבאה: בהינתן קלט x , הפעולה תחזיר את סכומם הכולל של כל האיברים בעץ הקטנים מ- x . הראו כיצד ניתן לממש את הפעולה הבאה בעץ חיפוש בינארי (מותר לשנות ולהוסיף נתונים לעץ). חישובו על הסיבוכיות המינימלית לפתרון הבעיה.

(ב) לכל קודקוד u בעץ בינארי T , נגדיר את הפונקציה $f(u)$ להיות אורך המסלול הקצר ביותר בין u לבין עלה כלשהו שהוא צאצא של u . נתבונן בעץ *AVL*. בנוסף לפעולות הרגילות, נרצה לתמוך בפעולה הבאה - לכל קודקוד u כקלט, נרצה לחשב את $f(u)$. הראו כיצד לממש פעולה זו בצורה יעילה ב-*AVL*. מותר להוסיף מידע נוסף על קודקודי העץ, ולשם כך הראו כיצד משנים את *insert* ו- *delete* בהתאם.

שאלה 5. חשבו על מבנה נתונים מתאים לבעיה הבאה:

- *access* - גישה לאלמנט ה- i במבנה הנתונים תעשה ב- $O(1)$.
 - *insert* - הכנסה תעשה לפי סדר. אם במבנה הנתונים יש כרגע j אלמנטים, ההכנסה תעשה למקום $j + 1$.
 - אין צורך לתמוך ב-*remove*.
 - מספר האלמנטים שמכניסים למבנה הנתונים אינו ידוע, ויכול לגדול עד אינסוף.
- המגבלה היחידה היא כמות הזיכרון הפנוי הלא מנוצל. נניח כי מותר להשתמש רק ב- k תאים לא מנוצלים.
- (א) הצג מבנה נתונים העומד בדרישות. הוכח: העלות לשיעורין של n פעולות *insert* הוא $O(n/k)$ לכל פעולה.
- (ב) יהי $c > 0$ קבוע כלשהו, ונניח כי מותר להשתמש ב- $c \cdot k$ כמות זיכרון לא מנוצל. הצג מבנה נתונים התומך בהכנסה בזמן $O(n/k^c)$.