מבני נתונים

תרגיל 11, סמסטר ב', תש"ע
Selection

בחירה

- נתון לנו אוסף של איברים לא ממוינים עם יחס סדר
- אנחנו רוצים למצוא את האיבר ה-i-י לפי הסדר (i th order statistic)
- בשיעור ראינו/נראה שני פתרונות לבעיה:
  - פתרון סטטי – אלגוריתם בחירה
  - פתרון דינמי – עץ חיפוש עם פעולת ()select – order statistics tree

תרגיל 1

- נתון מערך לא ממוין בגודל n. אנחנו רוצים למצוא את k האיברים הקטנים במערך בזמן $O(n)$
  - כבר ראינו פתרון ב-$O(n + k \log k)$
  - כבר ראינו חסם תחתון תואם אם אנחנו רוצים את האיברים ממוינים.

פתרון 1

- נמצא את האיבר ה-k בגודלו בזמן $O(n)$, נקרא לו x
- נעבור על המערך, וניקח כל איבר שקטן מ-x, בזמן $O(n)$
- נשלים את האיברים שלקחנו ל-k איברים ע"י בחירת איברים נוספים שווים ל-x

תרגיל 2

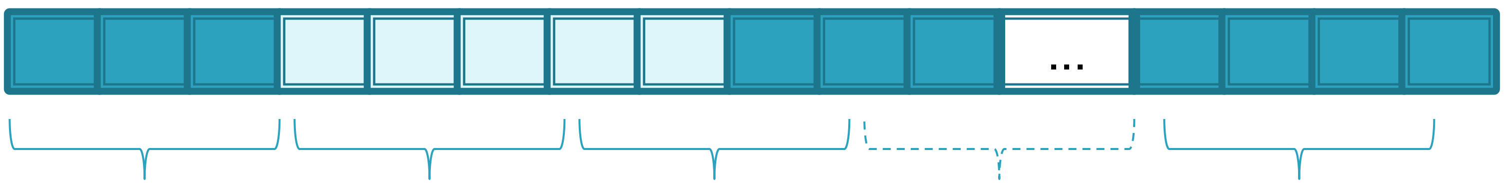
- נתון לנו מערך בגודל n של מספרים. אנחנו יודעים שקיים במערך איבר שחוזר לפחות $n/5$ פעמים. איך נמצא אותו בזמן לינארי?

רמז

- נניח שהמערך ממוין, איך נפתור את השאלה אז?

פתרון 2

- נבחר מספר קטן מ-$n/5$. למשל $x = n/6$
- האיבר שאנחנו מחפשים חייב להיות האיבר ה-x-י, או ה-$2x$-י או ה-$3x$-י, או ה-$4x$-י, או ה-$5x$-י, או ה-$6x$-י



- סה"כ יש לנו מספר קבוע (6) של מעמדים
- נמצא את כולם, ונעבור על המערך כדי לבדוק מי מהם מופיע מספיק פעמים
- זמן הריצה הוא $O(n)$

תרגיל 3

- נתונים לנו שני מערכים ממוינים A ו-B בגודל n כל אחד המכילים מספרים שונים. כיצד נמצא את החציון של מיזוג המערכים?
  - אם נמזג את המערכים נקבל שוב מערך ממוין ונוכל למצוא את החציון בזמן קבוע. אבל עצם המיזוג דורש זמן לינארי ב-n.

פתרון 3

- נמצא בזמן קבוע את החציון של A, m_A ואת החציון של B, m_B.
נניח בלי הגבלת הכלליות $m_A < m_B$
- החציון המשותף הוא בין m_A ו-m_B
  - מכיוון שמספר האיברים שקטנים ממנו ומספר האיברים שגדולים ממנו הוא n או $n-1$
- החציון המשותף הוא גם החציון המשותף למחצית A שגדולה מ-m_A ולמחצית B שקטנה מ-m_B
  - כי אנחנו מוציאים מהמערכים אותה כמות של איברים שגדולה מהחציון וכמות של איברים שקטנה מהחציון
- נמשיך ברקורסיה עד שנשאר עם איבר בודד
- עומק הרקורסיה $O(\log n)$, זמן הריצה של כל שלב קבוע מכאן זמן הריצה הכולל לוגריתמי ב-n

תרגיל 4

- נתון מערך בגודל $n = 2^k$. אנחנו יודעים שקיים איבר שמופיע $n/2$ פעמים במערך, איבר שמופיע $n/4$ פעמים במערך, איבר שמופיע $n/8$ פעמים במערך, וכן הלאה
- כיצד נמיין את המערך באופן היעיל ביותר?
  - האם החסם התחתון $\Omega(n \log n)$ תקף?

פתרון 4

- במערך יש $O(\log n)$ ערכים שונים
- נמצא את האיבר הנפוץ ביותר בזמן לינארי, כמו בתרגיל 2, ונעבור על המערך כדי להוציא ממנו את כל העותקים של האיבר הנפוץ
- נחזור על הפעולה כל עוד יש איבר שמופיע יותר מפעם אחת
- כל פעם המערך קטן בחצי, ולכן זמן הריצה למציאת כל האיברים הוא $O(n)$
- נרכיב מערך בגודל $O(\log n)$ שכל איבר מופיע פעם אחת
- נמיין את המערך החדש בזמן $O(\log n \log\log n)$
- נכפיל בחזרה, בזמן לינארי, את האיברים שחוזרים על עצמם במערך הממוין

תרגיל 5

- תארו מבנה נתונים המתחזק קבוצה של נקודות במישור, כאשר כל נקודה p_i מיוצגת כזוג מספרים (x_i, y_i)
- מבנה הנתונים צריך לתמוך בפעולות insert, delete ו- find בזמן גרוע ביותר O(log n), וכמו כן תומך בפעולה $\text{CountPointsAtDistance}(d_1, d_2)$ המבצעת את הפעולה הבאה בזמן המהיר ביותר שתוכלו:
  - בהינתן שני מספרים d_1, d_2 מהו **מספר הנקודות** במבנה שמרחקן מהראשית גדול מ- d_1 וקטן מ- d_2
- נניח שאין במבנה שתי נקודות בעלות אותו מרחק מראשית הצירים

פתרון 5

- בהינתן נקודה p_i ניתן לחשב בזמן קבוע את המרחק שלה מראשית הצירים
- נשתמש ב-order statistics tree. המפתח של p_i בעץ יהיה המרחק של הנקודה מראשית הצירים
- פעולות insert, delete ו-find יבוצעו על ידי חישוב המרחק של הנקודה מראשית הצירים, וביצוע הפעולה על העץ
- פעולת CountPointAtDistance תתבצע ע"י מציאת הנקודה p הרחוקה ביותר מראשית הצירים שמרחקה קטן מ-d_2 והנקודה q הקרובה ביותר לראשית הצירים שמרחקה גדול מ-d_1.
התשובה היא ה-order statistic של p פחות ה- order statistic של q ועוד 1
- כל הפעולות בוצעו בזמן $O(\log n)$

תרגיל 6

- נגדיר פרמוטציה מסדר (n, m) באופן הבא:
- נסדר את המספרים 1 עד n במעגל ונצביע על 1
- כל עוד המעגל לא ריק, נתקדם m צעדים על האיברים שנותרו במעגל, נדפיס את האיבר שהגענו אליו ונוציא אותו מהמעגל (המצביע ישאר עליו, אבל יחזור למעגל מיד בצעד הבא)
  - איך נראית פרמוטציה מסדר (7,3)?
- תארו אלגוריתם יעיל ליצירת פרמוטציה מסדר (n, m)

פתרון 6

- נבנה order statistics tree שמכיל את המספרים 1 עד n
- נאתחל מצביע לערך 1
- כל עוד העץ לא ריק נבצע את הפעולות הבאות:
  - נסמן ב-i את ה-order statistics של האיבר שעליו המצביע, וב-k את מספר האיברים בעץ (מבנה הנתונים שלנו מאפשר את שתי השאילתות האלה)
  - נקדם את המצביע ל-$\text{select}((i + m) \bmod k)$
  - נמחק את האיבר שעליו המצביע, ולפני זה נזיז את המצביע לאיבר הקודם
- כל אחת מהפעולות בלולאה דורשות זמן $O(\log n)$, ומספר החזרות הוא n, לכן זמן הריצה הכולל הוא $O(n \log n)$