מבני נתונים 2021 סמסטר א' מועד ב'

שאלה 1

נשתמש בעץ AVL (עם שדה size שאפשר לתחזק באופן יעיל שהוא גודל תת העץ שהצומת הנוכחית היא השורש שלו) שמחזיק איברים שמייצגים את החדרים הפנויים. המפתח יהיה האינדקס של החדר.

$Init(n)$:

נאתחל את העץ עם המפתחות $\{1, \ldots, n\}$ בזמן $O(n)$ נעשה זאת באופן רקורסיבי, המפתח של השורש יהיה $n/2$ (עד כדי עיגול למעלה) ואז בזמן $O(1)$ נאחד את העצים הימני והשמאלי שבנינו באופן רקורסיבי. נוסחת הנסיגה תהיה הנוסחה הבאה

$T(n) = 2T(n/2) + O(1)$ שהפתרון שלה לפי שיטת המאסטר יהיה $T(n) = \Theta(n)$

$Find(i,j)$:

אם i בעץ נחזיר אותו, אחרת נוסיף את i ונשמור את העוקב שלו, נמחק את i ונחזיר את ונחזיר את העוקב אם הוא קטן מj ואחרת נחזיר שאין חדרים פנויים בתחום זה.

$Count(i,j)$:

נוסיף את i,j לעץ אם הם לא שם, נחשב $rank(i), rank(j)$ ונחזיר את ההפרש (במידה והוספנו את i,j נפחית בהתאם 0,1,2). נזכור למחוק את i,j מהעץ אם הוספנו אותם.

$Set(i,b)$:

אם $b = 1$ נוסיף לעץ צומת עם המפתח i ואם $b = 0$ נמחק מהעץ את הצומת עם המפתח i.

שאלה 2

א. ננתח את זמן הריצה של כל שלב מתוך ארבעת השלבים העיקריים של האלגוריתם ואז נחבר אותם ונשיג חסם עליון הדוק כמה שיותר:

-על מנת למצוא את האיבר השני/שלישי קטן ביותר של 6 איברים צריך $O(1)$ פעולות.
-כדי למצוא את חציון האיברים צריך לעשות $T(\frac{n}{6})$ פעולות.
-על מנת להשתמש בו כפיבוט צריך לעשות $O(n)$ השוואות.
-כדי להמשיך ברקורסיה צריך לכל היותר $T(\frac{5n}{6})$ פעולות שכן נפטרים מלכל הפחות- $\frac{n}{6}\cdot\frac{1}{2}\cdot 2=\frac{n}{6}$ איברים (הפיבוט במקרה גדול בדיוק מכל האיברים שהם שניים בגודלם בכל שישייה).

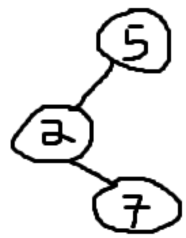
סה"כ נקבל שמתקיים $T(n)=O(n)+T\left(\frac{n}{6}\right)+T(\frac{5n}{6})$. הפתרון הוא $O(nlogn)$.

ב. נפעל בדיוק באותו אופן מקודם ואז נחשב את החסם העליון ההדוק ביותר:

-על מנת למצוא את האיבר השני/שלישי קטן ביותר של 6 איברים צריך $O(1)$ פעולות.
-כדי למצוא את חציון האיברים צריך לעשות $T(\frac{n}{6})$ פעולות.
-על מנת להשתמש בו כפיבוט צריך לעשות $O(n)$ השוואות.
-כדי להמשיך ברקורסיה צריך לכל היותר $T(\frac{29n}{36})$ פעולות שכן נפטרים מלכל הפחות- $\frac{n}{6}\cdot\frac{1}{3}\cdot 2+\frac{n}{6}\cdot\frac{1}{6}\cdot 3=\frac{7n}{36}$ איברים (השליש שהם שניים בגודלם ועוד שישית מאלו שלישיים בגודלם).
סה"כ נקבל שמתקיים $T(n)=O(n)+T\left(\frac{n}{6}\right)+T(\frac{27n}{36})$. הפתרון הוא $O(n)$.

<u>שאלה 3</u>

א. דוגמה לעץ שכזה:



ב. נניח בשלילה שניתן למיין עץ שכזה בזמן $o(nlogn)$. נראה כעת מיון השוואות שממיין n איברים בזמן $o(nlogn)$ וזו תהיה סתירה לחסם תחתון לבעיית המיון (באמצעות השוואות) שראינו בכיתה שאומרת שכל מיון מבוסס השוואות הוא $\Omega(nlogn)$. אכן בהינתן n איברים ניתן ליצור מהם עץ "נחמד" שהוא שרוך בסיבוכיות $O(n)$ ע"י סריקה לינארית של העץ, שבה בכל פעם משווים את האיבר הנוכחי לאיבר האחרון שהוכנס לעץ (האיבר שקודם לו במערך לדוגמה) והכנסה שלו במקום המתאים בעץ (נשמור גם מצביע בכל פעם לעלה שבקצה השרוך) כלומר אם הוא גדול יותר כבן ימני של העלה ואם לא כבן שמאלי ואז עדכון המצביע לעלה החדש. כעת אפשר למיין באופן הבא: ניצור מn איברים את אותו סוג עץ ב$O(n)$ ואז נמיין אותו ב$o(nlogn)$ (אפשר לעשות זאת לפי הנחת השלילה) ולכן קיבלנו אלגוריתם שממיין n איברים ב$o(nlogn) + O(n) = o(nlogn)$ וזו סתירה לחסם התחתון על בעיית המיון.

ג. נפעל לפי הרמז, נניח שיש לנו עץ "נחמד" מושלם כלומר בפרט $n = 2^h - 1$ עבור מספר $h \in \mathbb{N}$ כלשהו. היות ולא ניתן להסיק את יחס הסדר בין כל שני עלים ימניים (עלים שהם בנים ימניים של צומת כלשהיא) ויש $\frac{2^{h-1}}{2} = 2^{h-2}$ כאלו נסיק שבמיון מבוסס השוואות בהכרח נצטרך למיין את כל אותם עלים. לכן בעץ ההשוואות יהיו לפחות $2^{h-2}!$ עלים (אחד לכל פרמוטציה של אותם עלים). אבל נשים לב שהוכחנו בכיתה שהגובה של עץ עם k עלים הוא לפחות $logk$. לכן נקבל במקרה זה (הוכחנו בכיתה שמתקיימת הזהות $\log(n!) = \Theta(nlogn)$) שהגובה של עץ ההשוואות הוא לפחות -

$$\log(2^{h-2}!) = \Omega(2^{h-2}\log(2^{h-2})) = \Omega\left(\frac{n+1}{4}(\log(\mathrm{n}+1)-2)\right) = \Omega(\mathrm{nlogn})$$

ולכן בהכרח יש מסלול בעץ ההשוואות שהוא באורך $\Omega(\mathrm{nlogn})$ כלומר לא קיים אלגוריתם שממיין עץ שכזה ב$o(nlogn)$.

שאלה 4

א. נשתמש בטבלת hash בגודל n , המפתח יהיה זוג מספרים (i,j) שהוא המיקום של התא במישור ואם איבר אכן נמצא בטבלה אז זה אומר שהתא במערך תפוס ע"י הנחש במישור. בנוסף נשמור מערך מעגלי באורך n אשר שומר את המיקום של כל תא שהנחש נמצא בו כרגע, האיבר הראשון במערך תמיד יהיה המיקום של זנב הנחש. נשים לב שהאתחול יעלה $O(n)$ בתוחלת כי יש n איברים לעדכן (נכניס לטבלה ולמערך את הזוגות המתאימים). נתאר איך לבצע עדכון של הנחש ב$O(1)$ בתוחלת (כולל טיפול בהתנגשות):

עדכון הנחש:

ראשית נבדוק האם המקום שרוצים ללכת אליו מקיים ש$|x| = M \vee |y| = M$ ואם זה המצב נסיים את המשחק. אחרת נבדוק ב$O(1+\alpha) = O(2) = O(1)$ בתוחלת (כי מתקיים $\alpha = \frac{n}{m} = \frac{n}{n} = 1$ בכל רגע נתון) האם המקום שאליו רוצים ללכת כרגע תפוס ע"י הנחש, אם כן נבדוק האם זה הזנב (ב$O(1)$ כי זה האיבר הראשון במערך שלנו) ואם כן אז נעדכן את האיבר האחרון במערך להיות הזנב הנוכחי ואז נעדכן את האיבר הראשון החדש להיות האיבר השני הישן (במקרה זה אין צורך לעדכן את טבלת הhash). אם זה לא הזנב נסיים את המשחק. אם המקום לא תפוס אז נעדכן את טבלת הhash ע"י הוצאה של הזנב והכנסה של האיבר החדש, ואז נעדכן את האיבר האחרון להיות האיבר החדש ואת האיבר הראשון החדש במערך להיות האיבר השני הישן. כל זה ב$O(1)$ בתוחלת.

ב. הפעם נשתמש בעץ AVL בגודל n אשר מחזיק בצומת קואורדינטת x כלשהי ומצביע לעוד עץ AVL שבו כל קואורדינטות הy שמתאימות לאותו x שבהם הנחש נמצא. בנוסף נשמור כזוג מספרים את הקואורדינטה של הזנב של הנחש. נשים לב שהאתחול של המבנה יהיה ב$O(nlogn)$ כי צריך להכניס לעץ החיצוני n איברים ואז עוד n איברים לעצים פנימיים ולזכור מיהו הזנב הנוכחי) בנוסף נשמור מערך מעגלי כמו מקודם לתאים שתפוסים ע"י הנחש. נתאר איך לבצע עדכון של הנחש ב$O(logn)$ במקרה הגרוע (כולל טיפול בהתנגשות):

עדכון הנחש:

נבדוק האם האיבר החדש שרוצים להכניס מקיים $|x| = M \vee |y| = M$ ואם כן נסיים את המשחק. אחרת נבדוק ב$O(logn)$ האם המקום שרוצים ללכת אליו כרגע תפוס ע"י הנחש ע"י חיפוש ב$O(logn)$ של הx המתאים בעץ החיצוני ואז חיפוש בעוד $O(logn)$ בעץ הפנימי של אותה צומת של הy המתאים. אם הוא תפוס נבדוק האם זה הזנב ואם כן נוציא אותו ונכניס את האיבר החדש במקום ב$O(logn)$ סה"כ (כולל עדכון המערך כמו מקודם). אם זה לא הזנב נסיים את המשחק. אחרת אם האיבר לא נמצא אז נוסיף את האיבר החדש, נוציא את הזנב הנוכחי (ששוב – הוא האיבר הראשון במערך) ונעדכן את הזנב להיות האיבר הבא, ונכניס את האיבר החדש לעץ המתאים בתוך העץ החיצוני.

<u>שאלה 5</u>

א. הניסוח המחודש של הלמה:

Let x be a node of rank k and let $y_1, \ldots, y_k$ be the current children of x, in the order in which they were linked to x. Then, the rank of y_i is at least $i - 3$.

הוכחה: כאשר עשינו link לצומת הנוכחית וy_i היו לה $i - 1$ ילדים ולכן גם ל y_i (כי אפשר לעשות link רק לעצים בינומיים מאותה דרגה). מאז y_i איבדה לכל היותר 2 ילדים (אחרת היינו מנתקים אותה והיא הייתה הופכת לעץ עצמאי בערמה). לכן הדרגה שלה היא לכל הפחות $i - 3$.

ב. נשים לב שכעת אם יש צומת בדרגה k אז מספר הילדים שלה הוא :

$$S_k \geq S_0 + S_0 + S_0 + S_1 + \cdots + S_{k-3} = S_{k-1} + S_{k-3}$$

מנוסחת הנסיגה נסיק ש$S_k \geq c^k$ ולכן $n \geq S_k \geq c^k$ כלומר $logn \geq klogc \geq k$ לכן הדרגה חסומה ע"י $logn$.