מבני נתונים ואלגוריתמים 2 83224 – תרגיל 3 (תשפ"ד)

הנחיות הגשה:

- מועד הגשה: 26/9/2024, הגשה באתר מודל בלבד.
- ניתן להגיש בזוגות, על כל סטודנט <u>להגיש עותק משלו</u> באתר המודל.

שאלה 1:

שאלה זו עוסקת במבנה נתונים Union-Find שממומש ע"י "מערך של רשימות".

א. כתבו פסאודו קוד עבור פעולת UNION המקבלת שני איברים ומאחדת בין הקבוצות (הזרות) בהם הם נמצאים. דאגו שזמן הריצה יהיה סדר גודל של מספר האיברים בקבוצה הקטנה (כלומר הרשומות במערך יכילו גם שדה בשם size).

ב. הראו שלכל n קיימת סדרה של $n-1$ קריאות ל- UNION (לאחר שיצרנו n קבוצות בעזרת Make) שלוקחת זמן $\Theta(n\log n)$.

הסיקו שאנליזת סיבוכיות הפחת עבור המימוש הנ"ל הדוקה.

שאלה 2:

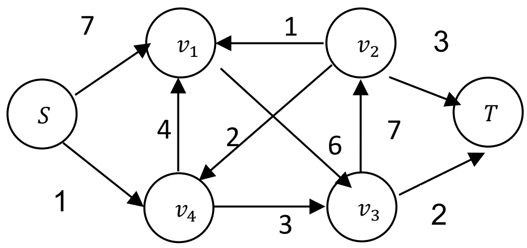
הוכיחו את משפט הסופרפוזיציה: תהי N רשת זרימה.

א) אם f_1 היא זרימה ברשת N, ו- f_2 היא זרימה ברשת N_{f_1} אז f_1+f_2 היא זרימה ברשת N שערכה $|f_1|+|f_2|$.

ב) אם f_1 ו- f_2 הן זרימות ברשת N, אז f_2-f_1 היא זרימה ברשת N_{f_1} שערכה $|f_2|-|f_1|$.

שאלה 3:

נתון גרף מכוון עם קיבולים על הקשתות:



א. חשבו את הזרימה המקסימלית ע"י Ford-Fulkerson.

ב. חשבו את הזרימה המקסימלית ע"י EK1.

ג. מהו חתך המינימום?

תארו שלבי ביניים והסבירו את תשובותיכם.

שאלה 4:

נתונה רשת זרימה בעלת זרימה חוקית f תארו אלגוריתם הבודק בזמן $O(|V|+|E|)$ האם הזרימה f מקסימלית או לא.

מבני נתונים ואלגוריתמים 2 (83224) - תשפ"ד
תקציר פתרון תרגיל בית 1

הערה: חלק מהפתרונות כוללים רק רעיונות מרכזיים, ולכן לא ניתן להתייחס אליהם כאל פתרון מלא.

תקציר פתרון לשאלה 1:

א. כאשר $a_i = 1$ לכל i, המערך נראה כך: [1 1 1 1 1 1 1 1 1 1].

במקרה זה כל המספרים במערך זהים. בכל חלוקה נוצרים שני תתי מערכים באורכים $n-1$ ו- 0.

ב. כאשר $a_i = i$ לכל i, המערך נראה כך: [1 2 3 4 5 6 7 8 9 10].

במקרה זה המערך כבר ממוין מקטן לגדול, ומכיוון שאנו בוחרים כ- pivot את האיבר האחרון במערך יוצא מצב זהה למקרה א', כשבכל חלוקה נוצרים שני תתי מערכים באורכים $n-1$ ו- 0.

ג. כאשר $a_i = (n-i)\, mod3$ לכל i, המערך נראה כך: [0,1,2,0,1,2,0,1,2,0].

בשלב הראשון יווצרו שני תתי מערכית אחד של אפסים והשני של 1 ו2. תת המערך של האפסים ימוין כמו בסעיף א, תת המערך של 1 ו2 ויחולק לשני תתי מערכים, אחד של 1 ואחד של 2. שניהם ימויינו כמו בסעיף א.

ד. כאשר $a_{2i+1} = i+5$ ו $a_{2i} = n-5-i$ לכל i, המערך נראה כך: [5,5,4,6,3,7,2,8,1,9].

מכיוון שבכל פעם ניקח בתור pivot את האיבר האחרון שהוא הגדול ביותר או הקטן ביותר במערך הנוכחי אנו נקבל כי בכל חלוקה ניצור שני מערכים בגודל $n-1$ ו- 0.

תקציר פתרון שאלה 2:

א. העומק המינימלי מקיים את המשוואה $n\alpha^h < 2$. לכן נקבל $h = \frac{1-\log n}{\log \alpha}$.

העומק המקסימלי מקיים את המשוואה $n(1-\alpha)^h < 2$. לכן נקבל $h = \frac{1-\log n}{\log(1-\alpha)}$.

ב. חלוקה שהיא מאוזנת לא פחות מחלוקה בפרופורציה α ו- $1-\alpha$ דורשת שהציר יהיה בתחום $[\lceil \alpha n \rceil, \lfloor (1-\alpha)n \rfloor]$. ההסתברות לבחור איבר בתחום היא בערך $1-2\alpha = \frac{n-\alpha n-\alpha n}{n}$.

תקציר פתרון לשאלה 3:

האלגוריתם:

- נמצא את ה-k-מחלק ה- $\lfloor (k-1)/2 \rfloor$ ע"י האלגוריתם למציאת סטטיסטי הסדר ה- i.
- כעת נחלק את המערך לשני חלקים ע"י שימוש באיבר ה- $\lfloor (k-1)/2 \rfloor$ כציר (pivot).
- בזאת חילקנו את הבעיה לשתי בעיות, ויש לפתור אותן באופן רקורסיבי.

ניתוח הסיבוכיות מסתמך על כך שעץ הרקורסיה הוא בעומק $O(\log k)$.

מבני נתונים ואלגוריתמים 2 (83224) – תשפ"ד
תרגיל בית 1

הנחיות הגשה:

- מועד הגשה: יום רביעי ה- 17/9/2024, הגשה באתר מודל בלבד.
- ניתן להגיש בזוגות, על כל סטודנט להגיש עותק משלו באתר המודל.

שאלה 1:

תארו את ריצת אלגוריתם Quicksort (הדטרמיניסטי) על הקלטים הבאים, כאשר $n = 10$:

א. $a_i = 1$ לכל i.

ב. $a_i = i$ לכל i.

ג. $a_i = (n - i)\ mod\ 3$ לכל i.

ד. $a_{2i-1} = i + 5$ ו- $a_{2i} = n - 5 - i$ לכל i, כאשר $1 \le i \le 5$.

שאלה 2:

יהי α קבוע שמקיים $0 \le \alpha \le \frac{1}{2}$.

א. הניחו שהחלוקה בכל רמה של Quicksort היא בפרופורציה α ו- $1 - \alpha$. הראו שהעומק המינימלי של עלה בעץ הרקורסיה הוא בערך $-\log n / \log \alpha$ ושהעומק המקסימלי של עלה בעץ הרקורסיה הוא בערך $-\log n / \log(1 - \alpha)$.

ב. הוכיחו שההסתברות ש- RandomizedPartition תייצר חלוקה שהיא מאוזנת לא פחות מחלוקה בפרופורציה α ו- $1 - \alpha$ היא בערך $1 - 2\alpha$.

הדרכה: התעלמו מעיגול מספרים.

שאלה 3:

נתון מערך A של n מספרים. ה-k-מחלקים של קבוצת מספרים בגודל n הם $k - 1$ האיברים שמחלקים את הקבוצה ל- k חלקים שווים בגודלם (עד כדי איבר אחד), בהנחה שהמספרים ממויינים. כלומר, מדובר בסטטיסטיים הסדר $\lfloor \frac{n}{k} \rfloor, \lfloor \frac{2n}{k} \rfloor, \dots, \lfloor \frac{(k-1)n}{k} \rfloor$.

תארו אלגוריתם שרץ בזמן $O(n \log k)$ שמוצא את ה-k-מחלקים של מערך נתון.
הוכיחו נכונות ונתחו את סיבוכיות האלגוריתם.

מבני נתונים ואלגוריתמים 2 (83224) – תשפ"ד
תקציר פתרון תרגיל בית 2

הערה: שימו לב שהפתרונות כוללים רק רעיונות מרכזיים, ולכן לא ניתן להתייחס אליהם כאל פתרון מלא.

תקציר פתרון לשאלה 1:

נחלק את סדרת הפעולות לפאזות שכוללת k פעולות מחסנית ופעולת גיבוי אחת. יש $\lceil m/(k+1)\rceil$ פאזות, כאשר האחרונה עשויה להיות פאזה חלקית שלא כוללת פעולת גיבוי. פעולות מחסנית לוקחות $O(1)$ זמן ופעולת גיבוי לוקחת $O(k)$ זמן. לכן סה"כ נקבל:

$$(m-\lfloor m/(k+1)\rfloor)\cdot O(1)+\lfloor m/(k+1)\rfloor\cdot O(k)=O(m)$$

ניתן לפתור בשיטת בחיובים: כל פעולת מחסנית משלמת מטבע אחד למימון פעולת ההעתקה העוקבת לה.
ניתן לפתור בשיטת האשראי: כל פעולת מחסנית משאירה מטבע אחד עבור פעולות העתקה.
ניתן לפתור ע"י פונקציית פוטנציאל: $\varphi=m \bmod (k+1)$.

תקציר פתרון לשאלה 2:

נשתמש ברשימה מקושרת עם שדה נוסף עבור size שיכיל את גודל הרשימה.

$\text{INSERT}(x,S)$ – מכניסים את x לראש הרשימה ומעדכנים את שדה size.

$\text{REMOVE-BOTTOM-QUARTER}(S)$ – מוצאים את סטטיסטי הסדר ה- $\left\lceil\frac{n}{4}\right\rceil$, ולאחר מכן עוברים על הרשימה ומוחקים ממנה את האיברים שקטנים/שווים לחציון.

ניתן להציג ניתוח דומה לדוגמא שהוצגה בתירגול ע"י שימוש בשיטת הפוטנציאל. ניתן גם להשתמש בשיטת החיובים, כאשר כל איבר שנכנס למחסנית מוסיף 4 מטבעות לקופה.

תקציר פתרון לשאלה 3:

א. העומק המקסימלי הוא $\lceil\log_{t_1} n\rceil$ והעומק המינימלי הוא $\log_{t_2} n$. ניתן לשים לב שלשורש יש מינימום שני ילדים ולכל שאר הצמתים יש מינימום t_1 ולכן $n\ge 2t_1^{h-1}$ כאשר h גובה העץ. בנוסף, יש מקסימום t_2 ילדים לכל צומת ולכן $n\le t_2^h$.

ב. הרחבה של פעולת MEMBER בעץ 2-3 כאשר בכל צמת עורכים חיפוש בינרי.
סיבוכיות: $O(\log_{t_1} n\cdot\log_2 t_2)=O(\log n)$ בהנחה שמתקיים $t_2/t_1=O(1)$.

נשים לב כשמדובר על עד t_2 ילדים לכל צומת – כלומר כעת בכל צומת שהוא לא עלה יש מערך שגודלו המקסימלי הוא t_2. זה אומר שאנחנו נצטרך לערוך חיפוש בינארי על המערך הזה בכל צומת שנגיע אליו כדי לדעת לאיזה ילד להמשיך. שימו לב שהרבה חילקו לשלושה תחומים ודבר זה לא נכון כי הגדלנו את כמות הילדים כדי שיהיו לנו תחומים קטנים יותר.

ג. פעולת INSERT מתרחבת באופן טבעי. התנאי שצריך להתקיים הוא: $t_2\ge 2t_1-1$ כי אחרת לא ניתן לפצל ולשמור על תכונת העץ. סיבוכיות: $O(t_2\log_{t_1} n)$. נשים לב שבמקרה זה הסיבוכיות היא $O(t_2\log_{t_1} n)$ ולא $O(\log_{t_1} n\cdot\log_2 t_2)$ כי הפעם בכל צומת שאנחנו עוברים דרכו לא רק שעושים חיפוש בינארי אלא הפעם גם יכול להיות שנצטרך לפצל צומת לשני צמתים וזה לוקח $O(t_2)$.

תקציר פתרון לשאלה 4:

אם $h_2 = h_1$ אזי ניצור צומת חדש שבו השורשים של T_1 T_2 יהיו הבנים שלו. נניח בה"כ כי $h_1 > h_2$ נתחיל בעץ T_1 ונטפס עד לרמה שבה גובה העץ השמאלי הוא h_2 ונוסיף את שורש העץ T_2 כבן של אותו הצומת אם לאותו הצומת היו 2 ילדים נוסיף את השורש של T_2 בתור הבן השמאלי ונזיז בהתאם אחרת נוסיף בתור הבן השמאלי ונאזן בהתאם כפי שלמדנו בכיתה.

מבני נתונים ואלגוריתמים 2 83224 – תרגיל 2 (תשפ"ד)

הנחיות הגשה:

- מועד הגשה: 19/9/2024, הגשה באתר מודל בלבד.
- ניתן להגיש בזוגות, על כל סטודנט <u>להגיש עותק משלו</u> באתר המודל.

שאלה 1:

נניח שאנחנו מבצעים סדרת פעולות על מחסנית שגודלה אף פעם לא עולה על k.
אחרי כל k פעולות מחסנית מערכת ההפעלה מעתיקה את תוכן המחסנית למטרת גיבוי.
סיבוכיות הזמן של פעולת ההעתקה היא O(l) בהנחה שיש l איברים במחסנית.
הוכיחו שסיבוכיות הזמן של כל סדרה של m פעולות, כולל פעולת העתקת תוכן המחסנית, היא O(m), בהנחה שבתחילת הסדרה המחסנית ריקה. כלומר יש להוכיח שסיבוכיות הפחת של הפעולות היא O(1).

שאלה 2:

תכננו מבנה נתונים שמתחזק קבוצה S של מספרים שלמים שונים שתומכת בשתי הפעולות הבאות:

- INSERT(x, S) – הכנס את האיבר x לקבוצה S.
- REMOVE-BOTTOM-QUARTER(S) – הוציאו את $\left\lceil \frac{n}{4} \right\rceil$ האיברים הקטנים ביותר מתוך S.

תארו מימוש של הפעולות בפסאודו-קוד כך שסיבוכיות הפחת של שתי הפעולות תהיה $O(1)$, כלומר שביצוע של סדרה של m פעולות תדרוש $O(m)$ זמן, בהנחה שבתחילת הסדרה המבנה ריק.

נתחו את סיבוכיות הפחת של המימוש שתיארתם.

שאלה 3:

מבנה נתונים של B-Tree הוא הכללה של עץ 2-3 שבו לכל צמת פנימי מלבד השורש יש לכל הפחות t_1 ילדים ולכל היותר t_2 ילדים. לשורש יש לפחות 2 ילדים ולכל היותר t_2 ילדים. (במקרה של עץ 2-3, $t_1 = 2$ ו- $t_2 = 3$.)
מלבד זאת, כמו בעץ 2-3, כל המספרים המוחזקים בעץ נמצאים בעלים, כאשר כל העלים במרחק זהה מהשורש, והמספרים בעלים מסודרים מקטן לגדול כאשר העלה השמאלי ביותר מחזיק את הערך המינימלי.

א. מה העומק המקסימלי האפשרי של עץ כזה אם יש לו n עלים?
מה העומק המינימלי?

ב. הציגו אלגוריתם לפעולת MEMBER באופן יעיל ככל האפשר.
נתחו את סיבוכיות האלגוריתם.

ג. הציגו אלגוריתם עבור פעולת INSERT.
איזה תנאי צריך שיתקיים על t_1 ו- t_2?
נתחו את סיבוכיות האלגוריתם.

שאלה 4:

נתונים שני עצי 2-3, T1 ו T2 כך שכל ה מפתחות בT1 קטנים מכל המפתחות ב -T2 . נסמן ב h1 וב h2 את הגובה של T1 ושל T2 בהתאמה. יש לציין שהגבהים יכולים להיות שונים .
תארו מימוש של הפעולה JOIN(T1,T2) שמחזירה עץ 2-3 שאיבריו הם האיחוד בין איברי T1 ו T2 כאשר סיבוכיות הזמן הנדרשת היא O(max(h1,h2)) הוכיחו נכונות ונתחו סיבוכיות

מבני נתונים ואלגוריתמים 2 (83224) – תשפ"ד
תקציר פתרון תרגיל בית 3

<u>הערה:</u> שימו לב שהפתרונות כוללים רק רעיונות מרכזיים, ולכן לא ניתן להתייחס אליהם כאל פתרון מלא.

תקציר פתרון לשאלה 1:

א. ה יש לשנות את הפסאודו-קוד שראיתם בהרצאה (שקף 14 בהרצאה 5) כך שהקבוצה הקטנה תתמזג לתוך הגדולה. יש לעשות זאת ע"י השוואת בשדה size של שני הנציגים. כמו כן, יש לעדכן את שדה הsize של נציג הקבוצה המאוחדת.

ב. נוכיח שלכל n, קיימת סדרת פעולות באורך n, כך שזמן הריצה של סדרת הפעולות הוא $\Theta(n \log n)$. הסדרה המתאימה לכך היא איחוד בודדים לזוגות, אח"כ זוגות לרביעיות, וכו'. עד שלבסוף מאחדים שתי קבוצות גדולות (ניתן לדמות זאת בעזרת עץ בינארי).

נניח ש- n הוא חזקה של 2, ונראה סדרה של $n-1$ קריאות UNION שדורשת זמן ריצה כולל של $\Theta(n \log n)$. הסדרה מחולקת ל- $\log n - 1$ תתי סדרות, כך שתת הסדרה ה- k מכילה את הקריאה $\mathrm{UNION}\left(2^k i + 1, 2^k i + 1 + 2^{k-1}\right)$, לכל $i \in \{0, \ldots, n/2^k - 1\}$.

ניתן להראות שזמן הריצה שדורשת כל תת סדרה הוא $\Theta(n)$.

תקציר פתרון לשאלה 2:

חוק הקשת:

$$f_1(u,v) + f_2(u,v) \le f_1(u,v) + c_{f_1}(u,v) = f_1(u,v) + c(u,v) - f_1(u,v) = c(u,v)$$

חוק הצמתי: $\sum_v [f_1(u,v) + f_2(u,v)] = \sum_v f_1(u,v) + \sum_v f_2(u,v) = 0$

סימטריה ניגודית: $f_1(u,v) + f_2(u,v) = -f_1(v,u) - f_2(v,u) = -[f_1(u,v) + f_2(u,v)]$

תקציר פתרון לשאלה 4:

נבנה רשת שיורית ונריץ BFS אם יש מסלול פשוט מs לt אזי ישנו מסלול שיפור ולכן הזרימה אין מקסימלית.

מבני נתונים ואלגוריתמים 2 83224 - תרגיל 4 (תשפ"ד)

הנחיות הגשה:

- מועד הגשה: יום רביעי ה- 31/7/24.
- ניתן להגיש בזוגות, על כל סטודנט להגיש עותק משלו באתר המודל.

שאלה 1:

ב*בעיית הגיבוי* נתון גרף דו-צדדי לא מכוון $G = (C, S, E)$, כאשר C היא קבוצת לקוחות ו- S היא קבוצת שרתים. לכל לקוח $c \in C$ יש קובץ שהוא רוצה לגבותו באחד משכניו השרתים (כל הקבצים באותו גודל), וכל שרת s יכול לאחסן עד $\ell(s)$ קבצים.

פתרון לבעיה הוא פונקציה $b: C \to S$ שמתאימה שרת לכל לקוח, כך שמתקיים:

- לקוח מקבל שירות רק משרת שכן, כלומר $(c, b(c)) \in E$, לכל לקוח c.
- העומס על כל שרת חסום ע"י מגבלת העומס שלו, כלומר $|\{c : b(c) = s\}| \leq \ell(s)$, לכל שרת s.

המטרה בבעיית הגיבוי היא למצוא פתרון שמביא למקסימום את מספר הלקוחות שקיבלו שירות.

א. הראו שבעיית שידוך המקסימום בגרף דו-צדדי היא מקרה פרטי של בעיית הגיבוי.

ב. תארו אלגוריתם יעיל שפותר את בעיית הגיבוי.
נתחו סיבוכיות והוכיחו נכונות.

שאלה 2:

השתמשו באלגוריתם ה- FFT לשם חישוב המכפלה של הפולינומים $A(x) = x$ ו- $B(x) = 1 + x$.

שאלה 3:

יש לכפול מטריצה A בת $2n$ שורות ו- n עמודות במטריצה B בעלת n שורות ו- $2n$ עמודות.

א. הסבירו כיצד ניתן לחשב את $C = AB$ ע"י שימוש באלגוריתם של Strassen.

ב. חשבו במדויק את מספר פעולות הכפל ומספר פעולות החיבור/חיסור הדרושות עבור $n = 2$.

שאלה 4:

בהינתן גרף לא מכוון $G = (V, E)$, קבוצה $I \subseteq V$ נקראת *בלתי תלויה* אם אין קשת בין כל שני צמתים מ- I, כלומר אם מתקיים $E \cap (I \times I) = \emptyset$. בבעיית הקבוצה הבלתי תלויה הקלט הוא גרף $G = (V, E)$, והמטרה היא למצוא קבוצה בלתי תלויה גדולה ביותר.

א. הוכיחו שהקבוצה I היא בלתי תלויה אם ורק אם הקבוצה $V \setminus I$ היא כיסוי בצמתים.

ב. תארו אלגוריתם יעיל, שבהינתן גרף $G = (V, E)$ שידוע שיש בו קבוצה בלתי תלויה שגודלה לפחות $\frac{3}{4}|V|$, מוצא בגרף קבוצה בלתי תלויה שגודלה לפחות $\frac{1}{2}|V|$.
נתחו סיבוכיות ויחס קירוב.

הדרכה: השתמשו באלגוריתם קירוב 2 עבור בעיית הכיסוי בצמתים.

שאלה 1:

בבעיית הגיבוי נתון גרף דו-צדדי לא מכוון $G=(C,S,E)$, כאשר C היא קבוצת לקוחות ו- S היא קבוצת שרתים. לכל לקוח $c\in C$ יש קובץ שהוא רוצה לגבות באחד משכניו השרתים (כל הקבצים באותו גודל), וכל שרת s יכול לאחסן עד $\ell(s)$ קבצים.

פתרון לבעיה הוא פונקציה $b: C\to S$ שמתאימה שרת לכל לקוח, כך שמתקיים:

- לקוח מקבל שירות רק משרת שכן, כלומר $(c,b(c))\in E$, לכל לקוח c.
- העומס על כל שרת חסום ע"י מגבלת העומס שלו, כלומר $|\{c: b(c)=s\}|\le \ell(s)$, לכל שרת s.

המטרה בבעיית הגיבוי היא למצוא פתרון שמביא למקסימום את מספר הלקוחות שקיבלו שירות.

א. הראו שבעיית שידוך המקסימום בגרף דו-צדדי היא מקרה פרטי של בעיית הגיבוי.

ב. תארו אלגוריתם יעיל שפותר את בעיית הגיבוי.
נתחו סיבוכיות והוכיחו נכונות.

1.א

בבעיית השידוך המקסימלי בגרף דו"צ $G=(L,R,E)$ אנו רוצים למצוא תת קבוצה של קשתות M זרות בצמתים כך שלכל צומת נוגע לכל היותר מקשת אחת ב-M ונרצה למקסם את הקבוצה M.

אם נגדיר בבעיית הגיבוי שלכל קובץ S $\ell(s)=1$ ולכן כל שרת S יכול לקבל קובץ יחיד, נקבל בדיוק בעיית השידוך המקסימלי.

כל צומת $s\in S$ מקביל לצומת $v\in R$ כאשר כיוון ש $\ell(s)=1$ צומת S יכול לגעת בקשת אחת בלבד אחרת יקבל 2 ומעלה קבצים וכך נחרוג מהמגבלה כי $|\{c:b(c)=s\}|=2\nleq 1$.

ולכן כיוון שבבעיית הגיבוי נרצה למצוא מקסימום לקוחות שיקבלו שירות וכל שרת יכול לקבל קובץ יחיד נרצה למקסם את כמות הקשתות הזרות בצמתים.

(נוכל לקבל קובץ מודד ולקוח c מצמה קובץ יחיד) וקיבלנו בעיה שקולה למציאת שידוך מקסימום.

1.ב

נשתמש באלגוריתם דומה לאלגוריתם למציאת שידוך מקסימלי רק שכעת קיבולן של הקשתות הנכנסות ל-T הן $\ell(s)$ לכל $s\in S$.

<u>האלגוריתם:</u>

(1) נתון גרף דו"צ לא מכוון $G(C,S,E)$ ונבנה רשת זרימה H כך:

א) נהפוך את הקשתות למכוונות מC לS וניתן לכל אחת מהן קיבול 1 כלומר $\forall u,v: c(u,v)=1$ — $O(|E|)$ מעבר על כל קשתות ב-G

ב) נוסיף מקור S ויעד T — $O(1)$

ג) נוסיף קשתות מכוונות מS לכל צומת בC בעלי קיבול 1 כלומר $\forall u: c(s,u)=1$ — $O(|C|)$ הוספת קשתות כמספר הלקוחות

ד) נוסיף קשתות מכוונות לT מכל צומת בS בעלי קיבולת $\ell(s)$ כלומר לכל $v\in S$ $c(v,t)=\ell(v)$ — $O(|S|)$ הוספת קשתות כגודל השרתים

(2) נמצא זרימת מקסימום ב H

Ford-Fulkerson: $O\big((\underbrace{|E|}_{|E|_H}+\underbrace{|V|}_{|S|+|C|})\cdot|V|\cdot\underbrace{\max\{\ell(s)\}}_{C_{max}}\big)$

$O(mF)=O(mnC_{max})$

(3) נבנה פונק' $b:C\to S$ כך שלכל $u\in C$ $b(u)=v$ עבור $v\in S$ המקיים $f(u,v)>0$ — $O(|E|)$ מעבר על כל הקשתות בין S לC ולקיחת אלו שגדולים מ0

המשך ↓

ד.2 המשך

ניתוח סיבוכיות: $O((|E|+|V|)\cdot|V|\cdot\max\{\ell(s)\})$ קיים גם מימוש של Edmond Karp

בסימון כיוון של $O(nm^2)$ ובמקרה שלנו $O(\underbrace{|V|}_{|S|+|C|+2}\cdot(\underbrace{|V|+|E|}_{E_H})^2)$

($+2$: s,t)

בגלל שאת ניתחתי בנפרד וסכמתי סיבוכיות
בסדר גודל של מציאת זרימה מקסימלית.

נכונות:

בהרצאה ראינו ניתוח נכונות לאלגוריתם שידוך מקסימלי. הניתוח דומה
וכן נראה רק את נכונות התוספת באלגוריתם,
וכי הפתרון b מחזיר את המבוקש.

תחילה נראה כי התנאים בשאלה מתקיימים:

* הגרף הוא דו"צ וכן לקוח מקבל שירות רק משרת וקיימת זרימה רק מלקוח לשירות

* כיוון שהקיבולות של הקשתות מ-$v\in S$ ל-T הינה $\ell(s)$ הזרימה המקסימלית הנכנסת
לכל צומת $v\in S$ היא $\ell(v)$ כי הזרימה לא מאפשרת יותר.

כעת נסביר למה הזרימה המקסימלית גוררת שבעצם ממקסמת מס' לקוחות
שקיבלו שירות.

נניח כי קיים $u\in C$ שלא קיבל שירות אז הפתרון האופטימלי b^*
ויקח u קיבל שירות רק ש $b(u)=v$ עבור $v\in S$

לפי ההפרדה שלנו b מתקיים $|\{c: b(c)=v\}|<\ell(v)$ אחרת לא היה
ניתן לספק שירות בפתרון האופטימלי דרך V.

אבל כיוון ש (1) גורר זרימה H שבעינו קיימת הזרימה לא רוויה
מ $e(u,v)$ שניתן להגדיל מזרימה מ-V ולכן הזרימה אינה מקסימלית

בסתירה לכך שהאלגוריתם מוצא זרימה מקסימלית גוררת

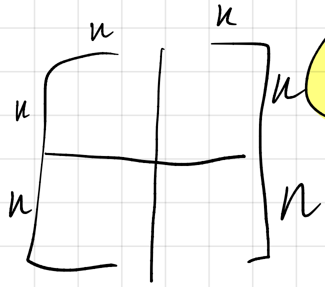
ולכן $b=b^*$

שאלה 3:

יש לכפול מטריצה A בת $2n$ שורות ו- n עמודות במטריצה B בעלת n שורות ו- $2n$ עמודות.

א. הסבירו כיצד ניתן לחשב את $C = AB$ ע"י שימוש באלגוריתם של Strassen.

ב. חשבו במדויק את מספר פעולות הכפל ומספר פעולות החיבור/חיסור הדרושות עבור $n = 2$.



א.3

$$A \in \mathbb{R}^{2n\times n} \quad B \in \mathbb{R}^{n\times 2n} \rightarrow AB \in \mathbb{R}^{2n\times 2n}$$

*כדי שאלגוריתם סטרסן יעבוד נניח ש n חזקה של 2.

אם n לא חזקה של 2 אפשר לרפד ב"0" ולהביא למימדים כך.

1. נפרק את A ו B לתתי מטריצות

$$A \rightarrow \begin{bmatrix} A_{11}, & A_{12} \\ A_{21}, & A_{22} \end{bmatrix} \qquad B \rightarrow \begin{bmatrix} B_{11}, & B_{12} \\ B_{21}, & B_{21} \end{bmatrix}$$

2. נחשב את תתי המטריצות הבאות:

$P_1 = A_{11}\cdot B_{12} - A_{11}\cdot B_{22}$ $\quad P_2 = A_{11}\cdot B_{22} + A_{12}\cdot B_{22}$ $\quad P_3 = A_{21}\cdot B_{11} + A_{22}\cdot B_{11}$

$P_4 = A_{22}\cdot B_{21} - A_{22}\cdot B_{11}$ $\quad P_5 = A_{11}\cdot B_{11} + A_{11}B_{22} + A_{22}B_{11} + A_{22}B_{22}$

$P_6 = A_{12}\cdot B_{21} + A_{12}B_{22} - A_{22}B_{21} - A_{22}B_{22}$ $\quad P_7 = A_{11}B_{11} + A_{11}B_{12} - A_{21}B_{11} - A_{21}B_{12}$

$C_{11} = P_5 + P_4 - P_2 + P_6$ $\quad C_{21} = P_3 + P_4$

$C_{12} = P_1 + P_2$ $\quad C_{22} = P_5 + P_1 - P_3 - P_7$

ולכן נקבל:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

ב.3

מתרגול עבור $n=2$ ראינו כי צריך 7 הכפלות ו 18 פעולות חיבור/חיסור

במקרה שלנו $n=4$ כלומר יש פי 4 איברים ולכן נצטרך פי 4 פעולות

כלומר:

פעולות כפל	$7\cdot 4 = 28$
פעולות חיבור/חיסור	$18\cdot 4 = 72$

שאלה 4:

בהינתן גרף לא מכוון $G=(V,E)$, קבוצה $I\subseteq V$ נקראת *בלתי תלויה* אם אין קשת בין כל שני צמתים מ- I, כלומר אם מתקיים $E\cap(I\times I)=\emptyset$. בבעיית הקבוצה הבלתי תלויה הקלט הוא גרף $G=(V,E)$, והמטרה היא למצוא קבוצה בלתי תלויה גדולה ביותר.

א. הוכיחו שהקבוצה I היא בלתי תלויה אם ורק אם הקבוצה $V\setminus I$ היא כיסוי בצמתים.

ב. תארו אלגוריתם יעיל, שבהינתן גרף $G=(V,E)$ שידוע שיש בו קבוצה בלתי תלויה שגודלה לפחות $\frac{3}{4}|V|$, מוצא בגרף קבוצה בלתי תלויה שגודלה לפחות $\frac{1}{2}|V|$.
נתחו סיבוכיות ויחס קירוב.

הדרכה: השתמשו באלגוריתם קירוב 2 עבור בעיית הכיסוי בצמתים.

א.4

כיוון א $\leftarrow$

נניח ש I בת"ל ונראה ש V/I כיסוי בצמתים

מהגדרה I בת"ל $\Leftarrow$ $\forall u,v\in I$ $\Rightarrow$ $\nexists(u,v)\in E$

- אם ל u יש צלע (u,k) אז $k\in V/I$ כי $k\notin I$ אך $k\in V$
- אם ל u אין צלע אז הוא קודקוד בודד ולכן לא משפיע על כיסוי הגרף ועדיין V/I מהווה כיסוי לגרף G.

מאחר ומנינו את כל האפשרויות ובכל אחת V/I כיסוי לגרף אז

I בת"ל $\Leftarrow$ V/I כיסוי לגרף

כיוון ב $\rightarrow$

נניח כי V/I כיסוי בגרף ונראה כי I בת"ל

V/I כיסוי לכן $\forall(u,v)\in E:(u\in V/I)\vee(v\in V/I)$

אם $u\in V/I$ אם $v\notin V/I$ (לפי עקרון הגבלת כלליות אותו דבר למקרה ההפוך):

אז מאחר ו $v\notin V/I$ אז $v\in I$ ומאחר ו $u\in V/I$ אז $u\notin I$
ולכן מתקיים ש I בת"ל אם רק מקרה זה מתקיים.

אם $u,v\in V/I$ אז $u,v\notin I$ לא סותר ש I בת"ל

אם $u,v\notin V/I$ מקרה זה לא אפשרי כי V/I כיסוי, ולפי הגדרת הצלע $(u,v)\in E$ u או v חייב להיות שייך V/I
מנינו כל האפשרויות ובכל אחת מהאפשרויות I קבוצה בת"ל
כלומר

I בת"ל $\longrightarrow$ V/I כיסוי לגרף

ומאחר והוכחנו 2 כיוונים אז:

I בת"ל $\Longleftrightarrow$ V/I כיסוי לגרף

שאלה 4:

בהינתן גרף לא מכוון $G=(V,E)$, קבוצה $I\subseteq V$ נקראת *בלתי תלויה* אם אין קשת בין כל שני צמתים מ- I, כלומר אם מתקיים $E\cap(I\times I)=\emptyset$. בבעיית הקבוצה הבלתי תלויה הקלט הוא גרף $G=(V,E)$, והמטרה היא למצוא קבוצה בלתי תלויה גדולה ביותר.

א. הוכיחו שהקבוצה I היא בלתי תלויה אם ורק אם הקבוצה $V\setminus I$ היא כיסוי בצמתים.

ב. תארו אלגוריתם יעיל, שבהינתן גרף $G=(V,E)$ שידוע שיש בו קבוצה בלתי תלויה שגודלה לפחות $\frac{3}{4}|V|$, מוצא בגרף קבוצה בלתי תלויה שגודלה לפחות $\frac{1}{2}|V|$.
נתחו סיבוכיות ויחס קירוב.

<u>הדרכה:</u> השתמשו באלגוריתם קירוב 2 עבור בעיית הכיסוי בצמתים.

ב.4

נמצא שידוך מקסימלי א על ידי אלגוריתם קירוב 2
הפלט של האלגוריתם הוא $V\setminus C$.

לפי סעיף א קודם $V\setminus C$ קבוצה ב"ת ומאחר וידוע
כי קיימת קבוצה ב"ת בגודל $\frac{3}{4}|V|$ לפחות אז

$$|C|\le\frac{1}{4}|V| \Leftarrow V\setminus C\le\frac{3}{4}|V|$$

$$\Downarrow$$

$$|C|\le\frac{1}{2}|V|$$

$$\Downarrow$$

$$\boxed{|V\setminus C|\ge\frac{1}{2}|V|}$$

שאלה 1:

בבעיית *הגיבוי* נתון גרף דו-צדדי לא מכוון $G = (C, S, E)$, כאשר C היא קבוצת לקוחות ו- S היא קבוצת שרתים. לכל לקוח $c \in C$ יש קובץ שהוא רוצה לגבותו באחד משכניו השרתים (כל הקבצים באותו גודל), וכל שרת s יכול לאחסן עד $\ell(s)$ קבצים.

פתרון לבעיה הוא פונקציה $b: C \to S$ שמתאימה שרת לכל לקוח, כך שמתקיים:

- לקוח מקבל שירות רק משרת שכן, כלומר $(c, b(c)) \in E$, לכל לקוח c.
- העומס על כל שרת חסום ע"י מגבלת העומס שלו, כלומר $|\{c : b(c) = s\}| \leq \ell(s)$, לכל שרת s.

המטרה בבעיית הגיבוי היא למצוא פתרון שמביא למקסימום את מספר הלקוחות שקיבלו שירות.

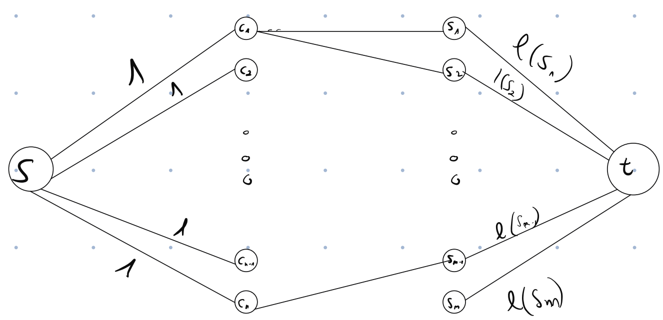
א. הראו שבעיית שידוך המקסימום בגרף דו-צדדי היא מקרה פרטי של בעיית הגיבוי.

ב. תארו אלגוריתם יעיל שפותר את בעיית הגיבוי.
נתחו סיבוכיות והוכיחו נכונות.

א) אפשר לראות שבעיה זו נובעת מבעיית השרתים ע"י עמידה בתנאים רק של אבל במשקל (לקוחות) יש התאמה

אחת בדיוק לצד ימין (שרתים). וצד ימין יכול לקבל כמות א שלו, [קיבולת של שרת היא $\ell(s) \geq 0$]

נבנה את הרשת:



ב) מכיוון שהמרחק הינו שווה כל מסלול (מרחק 3), אפשר להשתמש ב ford-fulkerson, שידוע שזמן הריצה שלו $O(E \cdot maxF)$.

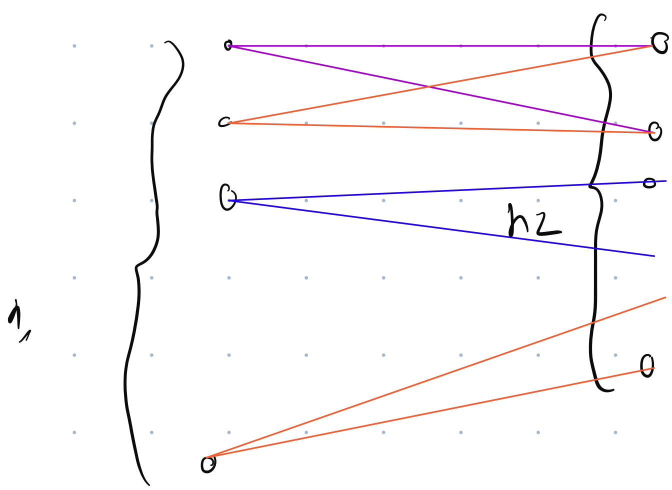
את הזרם המקסימלי יודעים לפי החתך המינימלי, שהוא מ $s \to c$ שהוא $|c|$ שהם n.

שאלה 2:

יהי $G=(L,R,E)$ גרף דו צדדי שבו דרגת כל צמת היא בדיוק d, עבור $d>0$.
שידוך מושלם הוא שידוך שבו כל צמת נוגע בקשת שידוך.

א. הוכיחו שמתקיים $|L|=|R|$.

ב. הוכיחו שקיים ב- G שידוך מושלם.
<u>הדרכה:</u> בנו רשת זרימה מתאימה, והראו קיום של זרימה שערכה $|L|$.



יהי G גרף שידוך דו צדדי כך ש

קיים $n_1=|L|$ ו $n_2=|R|$ שכל צומת דרגה d

כך ש $d>0$.

נניח ש $n_2 \neq n_1$.

אזי לכל $l\in L$ מחובר ל $\{r_1,\dots,r_d\}\in R$

שאלה 2

נגדיר את הקבוצה

$$O_r = \{\forall r \in R : f(r) \to \{e \in E \wedge \ell \in L : \exists e \in (r,\ell)\}\}$$

לכל איבר r נגדיר קבוצה של G השכנים שלו, בעצם יש לנו קבוצה של קבוצות.

$$O_r = |R|$$

מכיוון שבגלל של כל איבר r.

נגדיר באותו אופן את $O_L = |L|$.

אם נפתח את הקבוצה לפית קבוצה אחת נגדיר $T_r = \{\ldots O_r\}$

נקבל גודל קבוצה $|T_r| = d \cdot |O_r|$ מכיוון של r ב R אנחנו מייצרים בדיוק d קבוצות.

נראה $\forall \ell \in L$ מתקיים ℓ_i מופיע d פעמים בדיוק T_r.

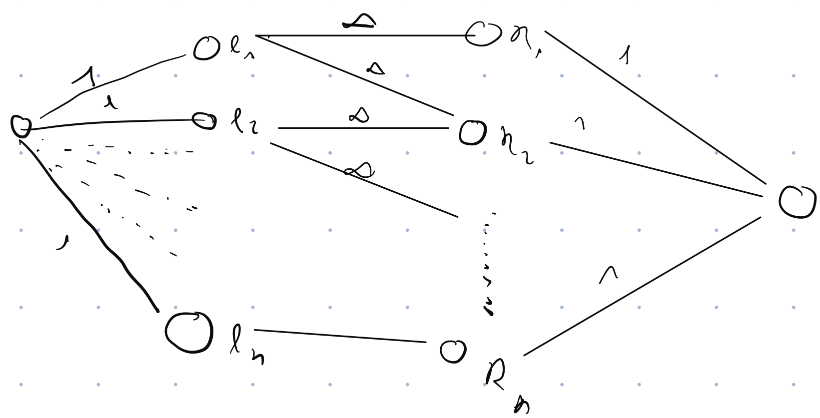
נניח בשלילה שקיים ℓ_i כך שהוא מופיע ב T_R כמות אשר שונה מ d, נגדיר כמות זו כ p.

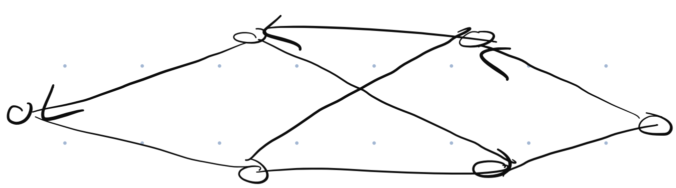
אם $p \neq d$ אזי הדרגה של G כמות היא לא d, בסתירה לנתון.

לכן אפשר לומר ש $|T_R| = d \cdot |L|$ שבעצם נקבל

$$|T_R| = d \cdot |O_R| = d \cdot |R| = d \cdot |L|$$

שאלה 2 סעיף ב.





נבנה רשת שיורית לאחר הורדה k מסלולים מכוונים
כלשהם

שאלה 3 ב

נשתמש במשפט החתונה על מנת להוכיח טענה, משפט החתונה אומר ש לגרף $G=(S,T,E)$, הוא גרף דו צדדי, קיים בו שידוך את S אם"ם לכל תת קבוצה $W \in S$ מתקיים ש $|W| \le |N(W)|$, אזי קיים התאמה מושלמת

נתון ש $d \ge 1$ ולכן $|W| \le d \cdot |W| \le |W|$ ולכן יש התאמה.

שאלה 3 סעיף א

נסביר כיצד ניתן לחשב את התוצאה של $C^{2n\times 2n} = A^{2n\times n} \cdot B^{n\times 2n}$ על ידי

אלגוריתם Strassen, כאשר נצטרך להשלים את שני המטריצות כך

שנוכל להשתמש באלגוריתם Strassen. נבצע זאת על ידי הוספת אפסים ליצור שני מטריצות

כי בנויים בגודל $2n\times 2n$.

לאחר מכן נמצא את $P_1 \dots P_7$, ונחשב את החישובים בצורה רקורסיבית.

$P_1 = (a+d)(e+h) = a\cdot e$

$P_2 = d(g-e) = 0$

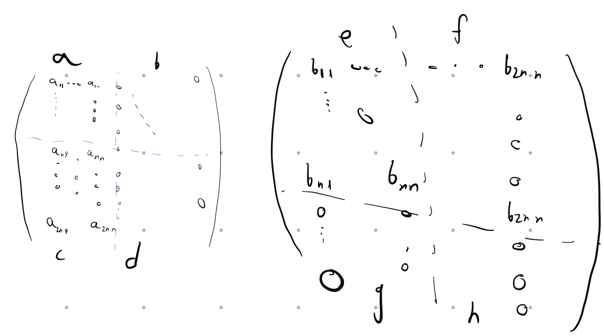
$P_3 = (a+b)h = 0$

$P_4 = (b-d)(g+h) = 0$

$P_5 = a(f-h) = a\cdot f$

$P_6 = (c+d)\cdot e = c\cdot e$

$P_7 = (a-c)(e+f)$



$\{b, d, g, h\} = 0$

שאלה 3 סעיף ב

$A^{4\times 2} \times B^{2\times 4} = C^{4\times 4}$

מה שרואים פה זה שלב הראשון של הרקורסיה נשתמש
במימדי המטריצה הידועים על מנת לחסוך בחישובים.
זה נותן לנו בצורה את הצורך לחשב רק $\{P_1, P_5, P_7\}$.
נעבור על כל אחד בצורה נפרדת על מנת לחשב זו.

$P_1 = (a+d)(e+h) = a\cdot e$

$P_2 = d(g-e) = 0$

$P_3 = (a+b)h = 0$

$P_4 = (b-d)(g+h) = 0$

$P_5 = a(f-h) = a\cdot f$

$P_6 = (c+d)\cdot e = c\cdot e$

$P_7 = (a-c)(e+f)$

$P_1 = a^{2\times 2} \times e^{2\times 2} = 8$ פעולות חישוב

$P_5 = a^{2\times 2} \times f^{2\times 2} = 8$ פעולות חישוב

$P_7 = a^{2\times 2}e^{2\times 2} + a^{2\times 2}f^{2\times 2} - c^{2\times 2}e^{2\times 2} - c^{2\times 2}f^{2\times 2} = 48$ פעולות חישוב

לאחר חישוב $P_1 \dots P_7$ כמות הפעולות הנוספות לחישוב המטריצה.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

$c_{11} = P_1 + P_2 - P_3 + P_4$

$c_{12} = P_5 + P_3$

$c_{21} = P_6 + P_2$

$c_{22} = P_5 + P_1 - P_6 - P_7$

סה"כ פעולות חישוב: 8

$P_7 = a^{2\times 2}e^{2\times 2} + a^{2\times 2}f^{2\times 2} - c^{2\times 2}e^{2\times 2} - c^{2\times 2}f^{2\times 2}$

$a^{2\times 2} e^{2\times 2} = 8$, $c^{2\times 2} e^{2\times 2} = 8$, $c^{2\times 2}\cdot f^{2\times 2} = 8$

$$\underbrace{\underbrace{(ae + af)}_{\text{8 פעולות}} + \underbrace{(-ce - cf)}_{\text{8 פעולות}}}_{\text{8 פעולות}} = 32 \text{ פעולות}$$

סה"כ כמות ב פעולות 64 פעולות

שאלה 4:

בשאלה זו נוסיף שדה שנקרא size לרשימת דילוגים. משמעות השדה היא שאם יש שני צמתים עוקבים a_1 ו- a_2 באותה רמה, אז השדה size של a_1 יכיל את מספר האיברים ברמה 0 שנמצאים בין a_1 ו- a_2.

א. תארו איך מתחזקים את ערך השדה בפעולות INSERT ו- DELETE.

ב. הסבירו איך ניתן למצוא את סטטיסטי הסדר ה- i ע"י שימוש בשדות ה- size שבמבנה.

נדבר על כל פעולה בנפרד,

כאשר מכניסים ערך אנחנו נרצה לבצע את הפעולות הבאות Inser -

```python
def insert(t, v, level):
    prevEl , nextEl = search(t, v)
    prevParentPath, distanceFromV = searchPath(t, prevEl)
    distanceToNext = prevParentPath.size
    remainingDistance = distanceToNext - distanceFromV
    prevEl.next = Node(v, prevEl.next)
    v.next = nextEl
    for path in prevParentPath:
        if path.dir != 'UP':
            continue
        if path.v <= v and path.next.v >= v:
            path.size += 1
    ADD_TO_TOP = random.choice([True, False])
    if ADD_TO_TOP:
        insert(t.levels[level + 1])

    return t
```

נדגיש שזה mockup

זה לא קוד אמיתי

אך הרעיון ברור

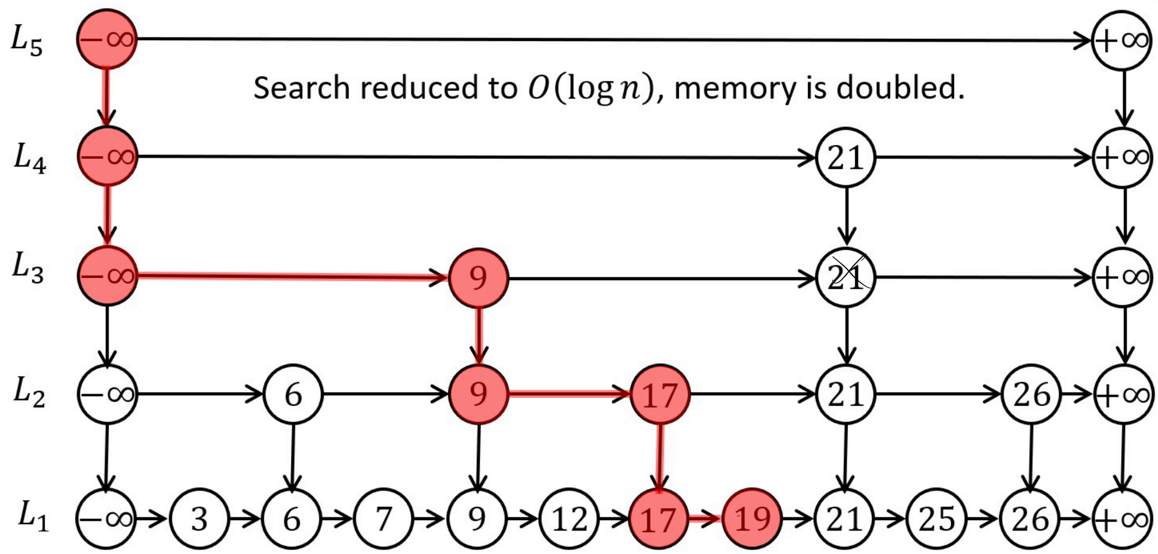
```python
def delete(t, v):
    prevEl , nextEl, didFound = search(t, v)
    prevParentPath, distanceFromV = searchPath(t, prevEl)
    if not didFound:
        prevEl.size - 1
    else:
        prevEl.size += v.size - 1

    prevEl.next = nextEl
    delete(t.levels[1] , v  )
```

כאשר נרצה לעשות מחיקה

נבצע רק סיבוב,

גם פה מדובר ב mockup.

```python
def findOrderStatistic(t, k, init):
    start = init
    end = init.next
    if start.size == k:
        return start
    if start.size > k and end.size < k:
        return findOrderStatistic(t, k , start)
    if start.size < k:
        return findOrderStatistic(t, k - start.size, end)
```


L_5
L_4
L_3
L_2
L_1
Search reduced to $O(\log n)$, memory is doubled.
$-\infty$
$+\infty$
3
6
7
9
12
17
19
21
25
26

מבני נתונים ואלגוריתמים 2 83224 – תרגיל 5 (תשפ"ד)

הנחיות הגשה:

- מועד הגשה: 4/5/2022, הגשה באתר מודל בלבד.
- ניתן להגיש בזוגות, על כל סטודנט להגיש עותק משלו באתר המודל.

שאלה 1:

כעת נחשב את הערכים הצפויים של C ו D-לפשטות, נניח שהמפתחות הם 1,2.....n. עבור שני צמתים שונים x,y נניח ש k = key[x] ו key[y] = i ונגדיר את משתנה האינדיקטור המקרי:

$$X_{i,k} = \begin{cases} 1 \; if \; y \; is \; a \; node \; on \; the \; right \; spine \; of \; the \; left \; subtree \; of \; x \; (in \; T) \\ 0 \; otherwise \end{cases}$$

הראו כי $X_{i,k} = 1$ אם ורק אם

priority[y]>priority[x]

key[y]<key[x]

כל z כך ש *key[y]<key[z]<key[x]* גורר *priority[y]<priority[z]*.

שאלה 2:

הוכיחו נכונות של דוגמא 2 בתרגול FFT

שאלה 3:

השתמשו באלגוריתם ה- FFT לשם חישוב המכפלה של הפולינומים $A(x) = 2 + x$ ו- $B(x) = 1 + x$.

שאלה 1:

כעת נחשב את הערכים הצפויים של C ו-D. לפשטות, נניח שהמפתחות הם 1,2,....n. עבור שני צמתים שונים y,x נניח ש key[y] = i ו k = key[x] ונגדיר את משתנה האינדיקטור המקרי:

$$X_{i,k} = \begin{cases} 1 \text{ if } y \text{ is a node on the right spine of the left subtree of } x \text{ (in } T) \\ 0 \text{ otherwise} \end{cases}$$

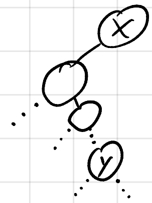
הראו כי $X_{i,k} = 1$ אם ורק אם

a) priority[y]>priority[x]

b) key[y]<key[x]

$h(y)>h(z)$

c) כל z כך ש key[y]<key[z]<key[x] גורר priority[y]<priority[z].

כלומר אם  אז $x_{i,k}=1$

הערה:

לפי מה שהבנתי ה- treap במקרה זה ערימת מינימלית

כלומר קודקוד עם p מינימלי הוא שורש וככל ש p קטן יותר הקודקוד גבוה יותר. (בהרצאה השתמשו במקסימלי)

צ"ל $X_{i,k}=1 \iff \{P[y]>P[x]$, $key(y)<key(x)$, ולכל z שמקיים $key(y)<key(z)<key(x)$ אז $P(y)<P(z)\}$

$\boxed{\text{כיוון א } \leftarrow X_{ik}=1}$

a) נניח ש $X_{ik}=1$ לכן y נמצא במורד שדרה הימנית של הבן השמאלי של x.

מכך נובע כי $h(x)>h(y)$ גובה של y קטן מגובה של x. לפי הגדרה של ערימת min, הגובה מציין את העדיפות לכן $P(x)<P(y)$ כי זו ערמת מינימום.

כלומר $P(y)>P(x)$. a ✓

b) מאחר ו y הוא נמצא בתת עץ שמאלי של x ומהגדרת עץ בינארי כל המפתחות קטנים ממפתח של x.

כלומר $key(y)<key(x)$ b ✓

c) יהי z כך ש $key(y)<key(z)<key(x)$

נמנה את האפשרויות:

מאחר ו $key(z)<key(x)$ אז z לא בן ימני של x והוא חייב להיות נכד/בן שמאלי של x.

ומאחר ו $key(y)<key(z)$ וגם y על במורד שדרה ימני של בן שמאלי של x אז גם מחויב ש z בן/נכד ימני של y אחרת יש סתירה בין 2 הנתונים.

מאחר ו z נכד/בן של y אז $h(y)>h(z) \Rightarrow P(y)<P(z)$ כנדרש.

$\boxed{\text{כיוון ב } a,b,c \rightarrow}$

מ a נובע כי $h(x)<h(x)$

מ b נובע כי y נכד/בן שמאלי של x

מ c נובע ש y גבוה מכל המפתחות שגדולים ממנו וקטנים מ x נקרא להם z,

כל המפתחות z הם נכדים שמאליים של x, מאחר וגדול מ-y הוא וגם בגובהה מהם אז y נמצא על קו ימני

שדרה ימני של בן שמאלי של x.

זה תואם להגדרה ש $X_{i,j}=1$ כנדרש.

שאלה 3:

השתמשו באלגוריתם ה- FFT לשם חישוב המכפלה של הפולינומים $A(x) = 2 + x$ ו- $B(x) = 1 + x$.

RECURSIVE−FFT$((a_0, a_1, \dots, a_{n-1}), n)$
- if $n == 1$ return a // recursion basis $y_0 = a_0$
- $\omega_n = e^{2\pi i/n}$; $\omega = 1$ // initialize complex root of 1
- $y^{[0]}$ = RECURSIVE−FFT$((a_0, a_2, \dots, a_{n-2}), n/2)$ // solve for $A^{[0]}$ coef.
- $y^{[1]}$ = RECURSIVE−FFT$((a_1, a_3, \dots, a_{n-1}), n/2)$ // solve for $A^{[1]}$ coef.
- for $k = 0$ to $n/2 - 1$ // combine $\mathrm{DFT}_{n/2}$ solutions
  - (4) $y_k = y_k^{[0]} + \omega y_k^{[1]}$ // combine for k power of nth root of 1
  - (5) $y_{k+n/2} = y_k^{[0]} - \omega y_k^{[1]}$ // combine for opposite root of 1
  - $\omega = \omega\omega_n$ // get next complex root of 1
- return y // return the $n/2$ point evaluations

$A(x) = (2,1,0,0)$ $\qquad B(x) = (1,1,0,0)$

$AB(x) = x^2+3x+2 \rightarrow (2,3,1,0)$

$FFT[(1,1,0,0),4]$

$y^{[0]} = FFT[(1,0),2]$ $\qquad y^{[1]} = FFT[(1,0),2]$

$y^{[0]} = 1$ $\quad y^{[1]} = 0$ $\qquad y^{[0]} = 1$ $\quad y^{[1]} = 0$

$y_0 = 1 + w\cdot 0$
$y_1 = 1 + w\cdot 0$
$(1,1)$

$y_0 = 1 + w\cdot 0$
$y_1 = 1 + w\cdot 0$
$(1,1)$

$y_0 = 1 + 1 = 2$
$y_2 = 1 - 1 = 0$

$y_1 = 1 + i$
$y_3 = 1 - i$

$(2, 1+i, 0, 1-i)$

$FFT[(2,1,0,0),4]$

$y^{[0]} = FFT[(2,0),2]$ $\qquad y^{[1]} = FFT[(1,0),2]$

$y^{[0]} = 2$ $\quad y^{[1]} = 0$ $\qquad y^{[0]} = 1$ $\quad y^{[1]} = 0$

$y_0 = 2 + 1\cdot 0 = 2$
$y_1 = 2 - 1\cdot 0 = 2$
$(2,2)$

$y_0 = 1 + w\cdot 0$
$y_1 = 1 + w\cdot 0$
$(1,1)$

$y_0 = 2 + 1 = 3$
$y_2 = 2 - 1 = 1$
$y_1 = 2 + i$
$y_3 = 2 - i$

$(3, 2+i, 1, 2-i)$

$(6, 1+3i, 0, 1-3i)$

$IFFT(6, 1+3i, 0, 1-3i)$ $\qquad \underset{0\ \ 1\ \ 2\ \ 3}{[2,3,1,0]}$

$y^{[0]} = IFFT(6,0) = (6,6)$ $\qquad y^{[1]} = IFFT(1+3i, 1-3i) = (2, 6i)$

$y_0 = (6+2):4 = 2$
$y_2 = (6-2):4 = 1$
$y_1 = (6 - i(6i)):4 = 3$
$y_3 = (6 + i(6i)):4 = 0$

$[2,3,1,0] \rightarrow x^2+3x+2 = (1+x)(x+2)$ ✓

שאלה 1:

הרצנו את הקוד בפיתון ורשמתי את הקוד והתוצאות.

https://github.com/1shaked/year_3/blob/main/algo2/HW1/quicksort.py

--------A--------

A: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

left: []

middle: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

right: []

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

--------B--------

B: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

left: [0, 1, 2, 3, 4]

middle: [5]

right: [6, 7, 8, 9]

left: [0, 1]

middle: [2]

right: [3, 4]

left: [0]

middle: [1]

right: []

left: [3]

middle: [4]

right: []

left: [6, 7]

middle: [8]

right: [9]

left: [6]

middle: [7]

right: []

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

--------C--------

C: [1, 0, 2, 1, 0, 2, 1, 0, 2, 1]

left: [1, 0, 1, 0, 1, 0, 1]

middle: [2, 2, 2]

right: []

left: []

middle: [0, 0, 0]

right: [1, 1, 1, 1]

left: []

middle: [1, 1, 1, 1]

right: []

[0, 0, 0, 1, 1, 1, 1, 2, 2, 2]

--------D--------

D: [5, 6, 3, 8, 1, 10, -1, 12, -3, 14]

left: [5, 6, 3, 8, 1, -1, -3]

middle: [10]

right: [12, 14]

left: [5, 6, 3, 1, -1, -3]

middle: [8]

right: []

left: [-1, -3]

middle: [1]

right: [5, 6, 3]

left: []

middle: [-3]

right: [-1]

left: [5, 3]

middle: [6]

right: []

left: []

middle: [3]

right: [5]

left: [12]

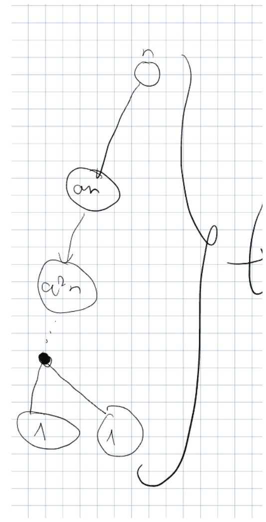
middle: [14]

right: []

[-3, -1, 1, 3, 5, 6, 8, 10, 12, 14]

שאלה 2:

אוקיי אז נוכיח, שאם אנחנו מניחים אורך t, ותיארנו באוויר את עץ הריקורסיה אזי מתקיים



$$a^t \cdot n = 1$$

$$\log_2 \textit{ on both sides} : \log_2(a^t \cdot n) = 0$$

$$\log a^t + \log n \ = 0$$

$$t \cdot \log a \ + \log n = 0$$

$$t = \frac{-\log n}{\log a}$$

עכשיו יש עלינו להבין למה זה גם האורך המינימלי,

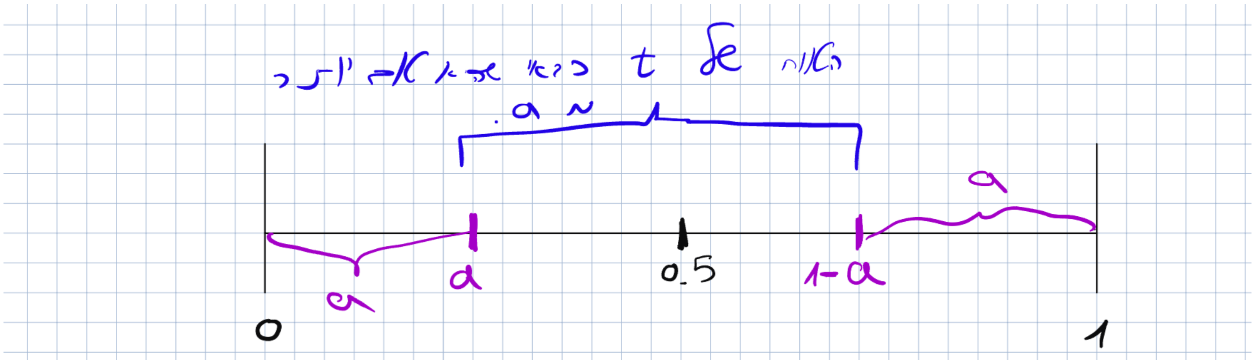
בעצם אנחנו בוחרים מספר שהוא קטן או שווה לחצי, שזה אומר שאנחנו מקטינים את הגודל של הרקורסיה. כמה שנקטין את המערך בכל שלב של הרקורסיה נצטרך פחות שלבים על מנת להגיע לאיבר אחד (שזה סוף הרקורסיה).

עכשיו אותו דבר נכון רק בצורה הפוכה לצד השני של הרקורסיה, שזה בעצם ככל שהצד השני של הרקורסיה גדול יותר ידרשו יותר שלבים על מנת להגיע לאיבר יחיד שהוא סוף הרקורסיה.

סעיף ב:

על מנת שהחלוקה תהיה פחות מאוזנת צריך שהממוצע של randomaztion partion ייתן בממוצע t כך ש $t<a$ או $t>1-a$

מה הסיכוי שבעצם החלוקה תהיה כזאת



אבל השטח הינו 1-2a שזה בעצם הסיכוי להיות הפרפור זה יהיה גבוה יותר.

מכיוון שהסיכוי הוא אחיד להיות בכל אחד מהסתברויות.

שאלה 3

בהינתן שנתון לנו k ו המערך באורך n.

מה שנצטרך לעשות זה למצוא את החלקים שעלינו לחלק את המערך אליו על מנת שנדע את ה k סטטיסטי שלו.

נבין איך לעשות את זה .

ראשית ניקח את אורך המערך n ונחלק אותו ב k (מניחים שהוא מתחלק שלם או קרוב לקח)

ניקח את הססטי האמצעי באותו מערך שהוא

$$a = \frac{k-1}{2} \cdot \frac{n}{k}$$

עכשיו ננסה למצוא את אותו סטטיסטי על a,

הוכחנו בכיתה שעל מנת למצוא סטטיסטי u צריך O(n)

עכשיו יהיה לנו מערך שיש לו את הססטיסטי ה a ממוין, שזה אומר שהוא נמצא במקום הנכון ואותו מצאנו, (כמובן נשמור את התוצאה)

שזה אומר המערך מחולק ל

$$\frac{a}{k} \cdot n \,, \left(1 - \frac{a}{k}\right) \cdot n$$

עכשיו נפעיל שוב את אותה הפונקציה שתיארנו פה רק שהפעם המערכים שנפעיל עליהם את הפונקציה יהיו בגדלים

$$a \,, n - a$$

והסטטיסטים שהם יתבקשו למצוא יהיו הרשימה של הסטטיסטים הקטנים מ a,

$$\frac{k-1}{2}$$

ואת הרשימה של הסטטיסטים הגדולים מ a,

$$\frac{k-1-1}{2}$$

הורדנו אקסטרה אחד מכיוון שאת אותו שסטטיסטי מצאנו בריצה זו שהוא האמצעי.

עכשיו ניקרא ברקורסיה לשני הצדדים של המערך, עם אותם דרישות.

זמן הריצה למציאה כל סטטיסטי יורד בערך בממוצע בחצי בכל ריצה.

״ אז מציאת הסטטיסטי הראשון לקח לנו O(n) זמן ריצה, אבל מציאת ברקורסיה את כל צד יקח לנו

$$O(a) + O(n-a) = O(n)$$

בממוצע מכיוון שחלקנו את המערך.

מוצאת כפול סטטיסטים ב O(n) זמן וככה ממשיך לאורך log(k) פעמים. מכיוון שכל פעם אנחנו מכפילים את כמות הסטטיסטים שמצאנו ב O(n) של זמן. ״

ולכן סה״כ זמן הריצה הינו

$$O(n \cdot \log(k))$$