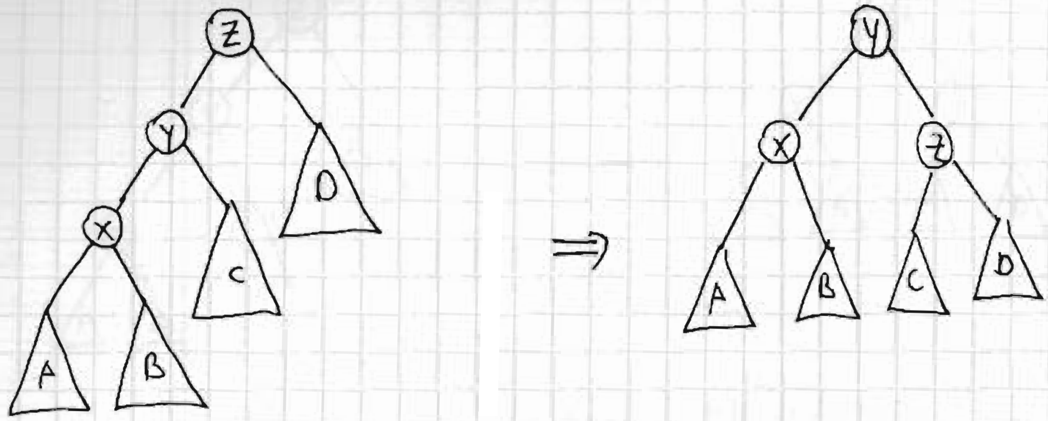
1. תחילה נסתכל על איך נראית הפעולה zig-zig

10/10



מכיוון ש zig ו zig-zag נשארו אותו דבר מספיק לבדוק את ה $am(zig\text{-}zig)$ נראה ש: $am(zig\text{-}zig) \leq 3(r'(y)-r(x))$

$$am(zig\text{-}zig) = ac(zig\text{-}zig) + \Delta\Phi = 1 + r'(x) + r'(y) + r'(z) - r(x) - r(y) - r(z)$$

$$\leq 1 + r'(y) + r'(z) - r(x) - r(z) \qquad // \; r'(x) - r(y) < 0$$

$$\leq 1 + r'(y) + r'(z) - r(x) - r(x) \qquad // \; r(z) - r(x) > 0$$

$$= 1 + r'(z) - r'(y) + r(x) - r'(y) + 3(r'(y) - r(x))$$

$$= 1 + \log\frac{s'(z)}{s'(y)} + \log\frac{s(x)}{s'(y)} + 3(r'(y) - r(x))$$

מכיוון ש $s'(z) + s(x) \leq s'(y)$ ותכונה ה log.

$$\leq 1 + \underbrace{\log\frac{s'(y)/2}{s'(y)}}_{=-1} + \underbrace{\log\frac{s'(y)/2}{s'(y)}}_{=-1} + 3(r'(y) - r(x))$$

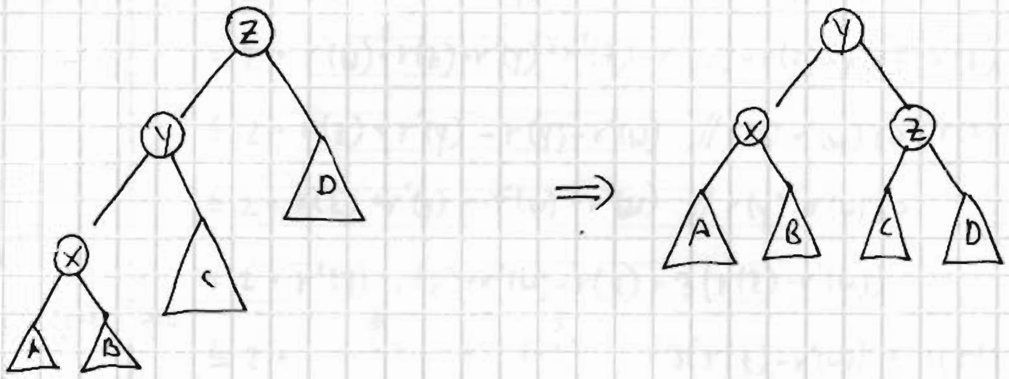
$$\leq 3(r'(y) - r(x))$$

סוכמים על כל הפעולות לכל המסלול

סכום של פעולות zig zig הוא אכן טלסקופי משום ש $r'(y)$ מקבל את ה $-r(x)$ של הפעולת zig zig הבאה (כי מסתכלים על אותו צומת x - החדש). ולכן ה amortized של כל פעולות zig zig: $3(r(t) - r(x))$

2. תחילה נטפל במקרה ה zig-zig top-down:

10/10



נראה ש $am(\text{zig-zig}) \le 3(r(z) - r'(x))$

$$am(\text{zig-zig}) = 1 + r'(x) + r'(y) + r'(z) - r(x) - r(y) - r(z)$$

$$= 1 + r'(x) + r(z) + r'(z) - r'(x) - r(y) - r(z) \quad // r(z) = r'(y),\ r'(x) = r(x)$$

$$\le 1 + r(z) + r'(z) - r'(x) - r(y) \quad // r(z) - r'(x) \ge 0$$

$$\le 1 + r(z) + r'(z) - r'(x) - r'(x) \quad // r(y) - r'(x) \ge 0$$

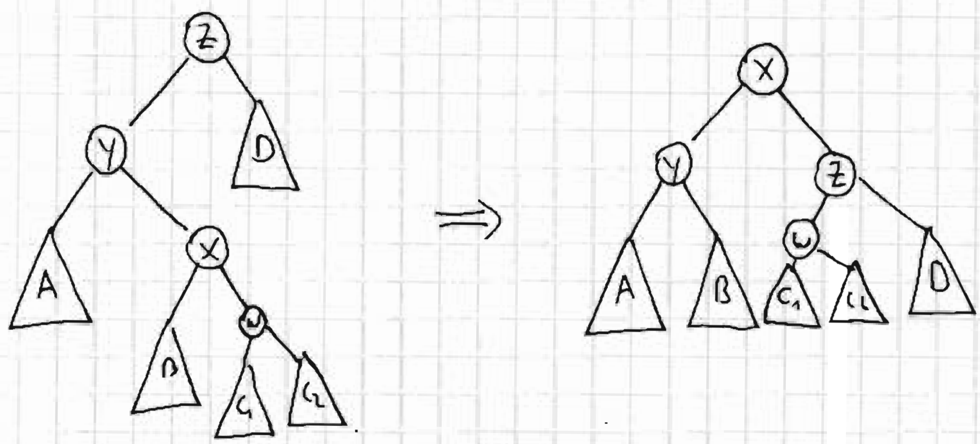
$$= 1 + r'(z) - r(z) + r'(x) - r(z) + 3(r(z) - r'(x))$$

אותו טיעון משאלה 1. (קמירות)

$$\le 1 + (-1) + (-1) + 3(r(z) - r'(x)) \le 3(r(z) - r'(x))$$

סכום פאזות אלו הוא טלסקופי מכיוון ש $-r'(x)$ מתבטל עם $r(z)$ משלב שמאחריו עד x שהופך ה-z הבא) סה"כ $\le 3(r(t) - r(x))$

נמשיך לטפל ב zig-zag top-down:



נראה e $am(zig\text{-}zag) \leq 3(r(z)-r'(w))$

$$am(zig\text{-}zag) = 2 + r'(w)+r'(x)+r'(y)+r'(z)-r(x)-r(y)-r(z)-r(w)$$

$$= 2 + r'(w)+r(z)+r'(y)+r'(z)-r'(w)-r(x)-r(y)-r(z) \quad // r'(x)=r(z),\ r(w)=r'(w)$$

$$\leq 2 + r(z)+r'(y)-r(y)-r'(w) \quad // r(x)-r'(w)\geq 0,\ r(z)-r'(z)\geq 0$$

$$\leq 2 + r(z)+r'(y)-r'(w)-r'(w) \quad // r(y)-r'(w)\geq 0$$

$$= 2 + r'(y)-r(z)+r'(w)-r(z)+3(r(z)-r'(w))$$

סוף איזון סדיק.

$$\leq 2 + (-1) + (-1) + 3(r(z)-r'(w)) \leq 3(r(z)-r'(w))$$

סכום פעולות אלו הוא טלסקופי מכיוון $-r'(w)$ מתבטל עם $r(z)$
של הפעולה הבאה (כי מעדכנים עם w שהופך ל z הבא) סה"כ $\leq 3(r(t)-r(x))$

את הלמה
לגבי הלוקליות
צריך להוכיח
כלל כל
הפעולות
יחד ולא לכל
אחת לחוד

.3

10/10

תחילה, נוסיף למבנה הנתונים שלנו ביט לכל צומת v, $r(v)$.

נסמן ב $R(v)$ את זוגיות סכום הצמתים ש $r(v)=1$ מ-v

עד לשורש דרך המסלול של v, .

בן שמאלי u של v הוא שמאלי באמת, אם:

- u הוא הבן השמאלי של v ו $R(v)=0$.
- u הוא הבן הימני של v ו $R(v)=1$.

אחרת u הוא הבן הימני האמיתי של v.

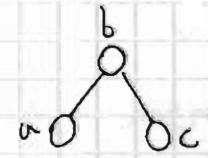
נוכיח שע"י היפוך הסימן $r(r)$ (של השורש). ה inorder של העץ

ששורשו r מתהפך:

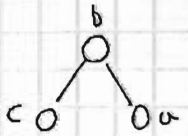
נוכיח באינדוקציה שהיפוך כל הבנים (ימני לשמאלי) יהפוך בעצם

את inorder בעץ:

<u>בסיס:</u> $h=2$



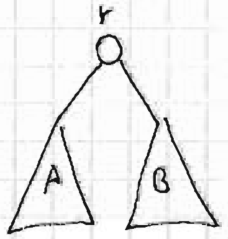
inorder: a, b, c



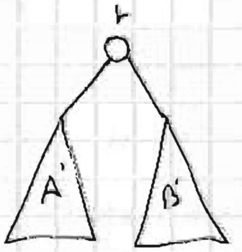
inorder: c, b, a

<u>הנחה:</u> נניח נכונות עבור עצים בגובה $\leq h$

<u>צעד:</u> נוכיח עבור $h+1$



inorder: $a_1, \dots, a_k, r, b_1, \dots, b_\ell$



inorder: $\underbrace{b_\ell, \dots, b_1}_{\text{לפי הנחת האינדוקציה}}, r, \underbrace{a_k, \dots, a_1}_{\text{לפי הנחת האינדוקציה}}$

הוכחנו שהיפוך כל הבנים [illegible] הופך את סדר בעץ.

נסיים בכך שאם נשנה את סימן $r(r)$ אזי $R(v)$ של כל הצמתים יתהפך,

ולכן כל הבנים השמאליים האמיתיים יהפכו לימניים אמיתיים ולהפך.

הפעולות rotate(v), splice(v) נשארות אותו דבר רק נוסף להן תנאי מקדים:

עבור rotate(v), כאשר u הוא האבא הסולידי של v נדרוש $r(v)=r(u)=0$

ועבור splice(v), כאשר u אביו נדרוש $r(u)=0$

נוסיף פעולה חדשה unreverse(v):

אם $r(v)=1$ אז:

$r(v) \leftarrow 0$

תחליף בין הבן השמאלי והימני של v

תשנה את הביט r של הבנים.

הפעולה splay(v) משתנה רק בכך שבתחילתה נדרש unreverse(x) לכל צומת x מהשורש עד ל-v.

כל שאר הפעולות לא משתנות (משתמשות בפעולות בפסאודו באותו אופן).

evert(v):

splay(v)

אם ל v יש בן ימני w:

החלף את הביט r(w)

אם ל v אין בן שמאלי אז w הבן השמאלי החדש

אחרת w יהפוך לבן שמאלי

אחרת v השורש האמיתי.

<u>נכונות:</u>

1. unreverse: שומרת על מבנה נכון של העץ עפ"י ההגדרה של בנים שמאליים וימניים אמיתיים.
2. evert: לאחר splay(v) v בשורש העץ הוירטואלי ולכן כל תת העץ הימני הסולידי שלו הוא המסלול בינו לבין השורש, שינוי r(w) דואג לכך (הוכחנו בתחילת השאלה). נותר רק "לתקן" את המסלול

תחת v. וזה נעשה ע"י הפיכה לבן שמאלי או ימני. אם

קיים כבר שמאלי.

3. שאר הפעולות שומרות על נכונות.

<u>סיבוכיות:</u>

1. unreverse(v) - $O(1)$

2. evert(v) - סיבוכיות splay(v) + $O(1)$

3. שאר הפעולות שומרות על הסיבוכיות כי לא שינינו אותן, למעט splay(v) שנוסף לה $O(1)$ לכל צומת מהשורש ל v שעולה גם ככה היא עושה. $O(1)$ לכל צומת כבר ולכן הסיבוכיות נשארת.

4. מבנה הנתונים שלנו יורכב מ Dynamic trees אשר יחזיקו עצים פורשים $\underbrace{\ }_{\text{מקסימליים}}$ של רכיבי הקשירות בגרף, תור FIFO שיחזיק את הקשתות ומונה C שייתן מספר לכל קשת שנכנסת ויאותחל ב-1.

10/10

<u>Connected(u,v)</u>

אם findroot(u) == findroot(v)

נחזיר true

אחרת נחזיר false

<u>insert(u,v)</u>

C = C+1

C עלול לחרוג בגודל מעבר לחסם ל-n

Q.insert(u,v,C)

evert(u)

אם connected(u,v) == true

נמחק מהגרף את הקשת המוקדמת ביותר:

$(w_1, w_2, c') = \text{mincost}(v)$

$\text{cut}(w_1)$

link(v,u,C)

<u>delete-oldest</u>

$(u, v, \tilde{c}) = Q.\text{pop}()$

evert(u)

$(w_1, w_2, c') = \text{mincost}(v)$

אם $c' \ne \tilde{c}$

לא נעשה כלום.

אחרת נבצע cut(v)

נכונות:

נוכיח את נכונות האלגוריתם, האינווריאנטה של dynamic trees תמיד שומרים יער פורש מקסימלי ואינווריאנטה ש Q מחזיק את כל הקשתות ביער ורק אותן.

מקרה בסיס: אין קשתות, Q ריק ויש n עצים שמחזיקים צומת יחיד.

נניח את נכונות האלגוריתם והאינווריאנטות עבור k פעולות.

צעד: הפעולה ה k+1:

1. connected(u,v):

יש מסלול בין u ל v ⇔ הם נמצאים באותו עץ פורש בגרף מקסימלי ⇔ לפי הנחת האינדוקציה u,v באותו עץ פורש במבנה ⇔ findroot(u), findroot(v) מחזירים את אותו שורש לפי נכונות dynamic trees.

האינווריאנטות נשמרות כי לא שינינו את המבנה.

2. insert(u,v):

אינווריאנטה 2 נשמרת, הכנסנו את הקשת ל Q.

נוסיף את הקשת (שיש לה משקל מקסימלי עד כה) ליער, אם סגרנו מעגל ⇔ connected(u,v) החזיר true

הקשת הקטנה ביותר במעגל זה היא המסלול בין u ל v ולכן ע"י כך שהפכנו את u לשורש evert(u) ומחקנו את הקשת mincost(v) (הקשת המינימלית בין v לשורש החדש u), אינווריאנטה 1 נשמרת.

נכונות הפעולה נובעת מאינווריאנטה 2.

3. delete-oldest():

אינווריאנטה 2 נשמרת, הוצאנו מ-Q את הקשת הקטנה ביותר

אינווריאנטה 1 נשמרת, הקשת הקטנה ביותר (u,v) נמצאת ב dynamic tree ⇔ היא הקשת הקטנה ביותר במסלול בין u ל v.

ע"י הפיכת u לשורש evert(u) והמציאה mincost(v) מצאנו את הקשת הקטנה ביותר במסלול בין u ל v. אם היא אכן שווה לקשת הקטנה ביותר בגרף אזי נמחק אותה מהיער.

נכונות הפעולה נובעת מהאינווריאנטות.

משקל
* נשים לב שלכל קשת יש מספר יחודי.

אז: הוכחנו את נכונות האלגוריתם.

<u>סיבוכיות:</u>

בכל פעולה עשינו $O(n)$ עבודה ו $O(n)$ קריאות לפעולות של dynamic trees

ולכן סיבוכיות ה amortized של פעולות ב dynamic trees:

סה"כ $O(\log n)$ לפעולה.

5. א. קשת (u,v) היא admissible רק אם $d(u) = d(v)+1$

לכן סידור הצמתים בסדר לא עולה של distance labels

יוצר מיון טופולוגי כאשר קשתות admissible הולכות רק קדימה.

pushes נעשים רק בקשתות admissible ולכן [redacted]

במהלך pass צומת יכול להפוך ל nonactive רק פעם אחת.

צומת הופך ל nonactive $\Leftrightarrow$ יש ממנו non saturating push.

לכן מס' non saturating pushes במהלך pass $\leq n$

[redacted] מספר ה [redacted] passes קטן מ $O(n^2)$, משום

שבכל מעבר חייב להיות לפחות relabel $\leq O(n^2)$ לאורך האלגוריתם

עובד.

$\therefore$ סה"כ מס' ה non saturating pushes $\leq O(n)\cdot O(n^2) = O(n^3)$.

10/10

5. ב. נשמור רשימה מקושרת ממוינת לפי distance label של כל הצמתים.

<u>אלגוריתם:</u>

1. נכניס לרשימה את G הצמתים בסדר G שבו.
2. כל עוד היו relabels במעבר על הרשימה, נעבור על הרשימה מהתחלה:
3. אם צומת אקטיבי:
4. נעשה לו discharge
   אם עשינו לו relabel נמצא את מקומו החדש לפי מיון הרשימה.
5. אחרת נעבור הלאה.
6. נחזיר את הזרימה שמצאנו.

<u>נכונות:</u>

1. הרשימה נשמרת ממוינת, שינינו distance label רק לצומת אחד לכל היותר במעבר 4 ומצאנו את מקומו מחדש.
2. הזרימה שנחזיר חוקית, אם במעבר על הרשימה לא עשינו relabel אזי הפסקנו לעשות discharge לכל הצמתים האקטיביים כי הם הפכו ללא אקטיביים, אזי אין excess בגרף ולכן הזרימה חוקית.
3. האלגוריתם עושה discharge לצמתים אקטיביים בלבד, כפי שלמדנו בכיתה והוכחת הנכונות תקיפה.

<u>סיבוכיות:</u>

1. צמתים לא אקטיביים: בכל מעבר (passes) שאנחנו עוברים על הרשימה לכל היותר פעם אחת, יש $O(n^2)$ passes (הוכחנו בכיתה) ולכן כמות הזמן שמשקיעים בצמתים לא אקטיביים $\leq O(n)O(n^2) = O(n^3)$
2. relabels: בכל relabel אנחנו מעבירים את הצומת למקומו החדש בעלות $O(n)$, אך יש לכל היותר $O(n^2)$ relabels ולכן סה"כ $\leq O(n)\cdot O(n^2) = O(n^3)$

ד. pushes: עפ"י הניתוח בכיתה של מבנה ה current edge

הזמן הכולל של discharge (ללא relabels) הוא

$$O(n \cdot m) + \#\text{of pushes} \le O(n^3) + \#\text{of pushes}$$

- saturating pushes: כראינו בכיתה [illegible] $O(n \cdot m) \le O(n^3)$
- non saturating pushes: הוכחנו ב-א $O(n^3)$