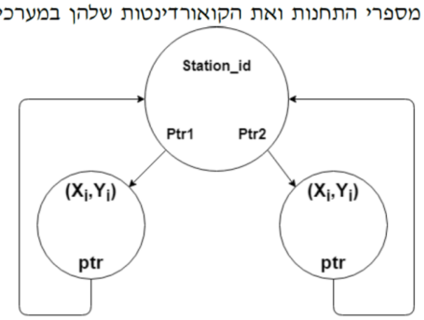
1) 1בשאלה 1 בסעיף 1 ממש בהתחלה, לא ברור מה זה מיון משני (לי זה ברור, אבל למתרגלים לא בטוח).

2) את כל הדבר הזה לא הבנתי בכלל:

3. 2 מבני $Master/Close$ במימוש שראינו בתרגול, שנכנה "$MCEast$" ו"$MCNorth$" בהתאמה. במבנים אלה נשתמש על מנת לייצג עבור כל תחנה את התחנה הפעילה הקרובה אליה ביותר לכיוון מזרח או צפון בהתאמה.

את מספרי התחנות ואת הקואורדינטות שלהן במערכים נייצג באופן הבא:




בצורה זו, במערך $Stations$ יאוחסנו תחנות, ומכל תחנה נוכל להגיע באמצעות מצביעים לאובייקט קואורדינטות שנמצא במערך $WestToEast$ או $SouthToNorth$, שמכיל בתורו מצביע בחזרה לאובייקט התחנה במערך $Stations$. בצורה זו מתאפשרת לנו "המרה" ב־$O(1)$ מייצוג אחד של תחנה לייצוג אחר במערך נפרד, ונראה בהמשך היכן תכונה זו רלוונטית.

מה ניסית להראות ולהסביר פה? מה זה Station_id? איפה המערכים בתמונה לא כל כך הבנתי..

3) ב-init ציינת שאנחנו משתמשים ב-bucketsort ושהוא אלגוריתם מיון יציב. זה לא נכון, כי bucketsort הוא לא יציב, ולאלגוריתם שלמדנו קוראים counting sort. הם משתמשים באותו עקרון בסיס – לשמור מערך שסופר, אך counting sort עובד בצורה שונה שגורם לו להיות יציב.

4) בכללי כל הפונקציה init לא ברורה לי כל כך. אולי כדאי שתקרא אותה מחדש ותנסח את הדברים שוב? יכול להיות שתקרא את זה פתאום תראה שפספסת דברים.

5) מה לגבי זה?

נמצא במערך זה את הקואורדינטות המתאימות לתחנה $end$ (בסיבוכיות $O(n)$ אורך המערך), ונתחיל מהתחנה הנמצאת במערך זה לפני $end$ נלך בכל איטרציה **אחורנית** עד שנגיע לתא בעל הקואורדינטות של התחנה $start$ (אנו יודעים לבצע המרה ב־$O(1)$ מאובייקט קואורדינטות לאובייקט האוגר מספר תחנה כאמור בתחילת הפתרון).

נאמר שתחנה היא "חוקית" אם היא קיימת (כלומר לא התחנה $n+1$), וכן קיים ממנה מסלול ל־$end$ (כלומר במערך $PathsArray$ הערך המספרי בה הוא אינו $-1$).

יש בטוח תחנה כזאת? אם כן, למה? אם לא, צריך לתקן.

6) פה צריך להסביר יותר טוב

במקרה זה, נשמור ב־$PathsArray$ באינדקס התחנה הנוכחית את אורך המסלול הקיים ב־$PathArray$ באינדקס התחנה החוקית שמצאנו, מוגדל ב־1 (שכן המסלול מהתחנה הנוכחית ל־$end$ ארוך בתחנה אחת מהמסלול שמתחיל מהתחנה הפעילה הבאה אחריה).
אם שתי התחנות שמצאנו חוקיות, נשווה בין אורכי המסלולים ששמורים בהן.
עם התחנה שהמסלול ממנה ארוך יותר נבצע את אותן פעולות שאנו מבצעים במקרה הקודם (נאחסן את אורך המסלול ממנה ל־$end$ בהוספת 1 בתא הנוכחי).

שאתה אומר "אינדקס התחנה החוקית שמצאנו" לקח לי זמן להבין למה התכוונת. צריך להדגיש שמדובר באינדקס של התחנה שקיבלנו חזרה מ-driveindirection.

7) גם פה:

אם שתי התחנות שמצאנו חוקיות, נשווה בין אורכי המסלולים ששמורים בהן.
עם התחנה שהמסלול ממנה ארוך יותר נבצע את אותן פעולות שאנו מבצעים במקרה הקודם (נאחסן את אורך המסלול ממנה ל־$end$ בהוספת 1 בתא הנוכחי).

אורכי המסלולים ששמורים בהן זה ניסוח קצת מטעה. הכוונה היא לאורכי המסלולים המקסימליים השמורים במערך patharray באינדקסים המייצגים את 2 התחנות החוקיות שמצאנו.

8) קצת מוזר הניסוח ולא הכי ברור:

מכיוון שאיטרציה שאנו מבצעים היא בסדר הפוך ממיקום התחנות שלהן ב־$WestToEast$, ובגלל המיון של $WestToEast$, נקבל שכאשר אנו מבצעים את הפעולות שהגדרנו עבור תחנה נתונה, כל תחנה הנמצאת מזרחית או צפונית לה על המפה היא כבר בעלת ערך מעודכן במערך $PathsArray$ (שכן היא נמצאת לאחר התחנה הנוכחית ב־$WestToEast$).
לכן נקבל שכאשר אנו מסיימים לבצע את הפעולות שהגדרנו על התא ה־$start$ במערך $PathsArray$, נקבל בתא זה את אורך המסלול הארוך ביותר מ־$start$ ל־$end$, ואת אורך זה נחזיר למשתמש.

9) כדאי אולי לשנות פה:

כלומר בסך הכול אנו מבצעים לכל היותר $2n$ פעולות $DriveInDirection$ שהסיבוכיות המשוערכת שלהן היא $O(\log^*(n))$, ולכן לפי הגדרת סיבוכיות משוערכת נקבל שאנו עושים זאת בסיבוכיות $2n \cdot \log^*(n) = O(n \cdot \log^*(n))$ **במקרה הגרוע**.
ולכן בסך הכול זו גם הסיבוכיות הכוללת של הפעולה.

לפי הגדרת הסיבוכיות המשוערכת, אם הסיבוכיות המשוערכת היא $O(h(n))$, אז לכל $n$ פעולות הסיבוכיות במקרה הגרוע היא $nh(n)$.

10) אינו*

אחרת, נכניס את $m+1$ מימין לעלה הימני ביותר הנוכחי ב־$H_1$ (נוכל לעשות זאת אם $H_1$ אכן אינה עץ שלם), באותה רמה.

11) חדשה*

כלומר כל האיברים פרט ל־$m+1$ מקיימים את הטענה מהנחת האינדוקציה.
נותר להראות ש־$m+1$ מקיים את התנאי:
אם בהכנסת $m+1$ הוספנו רמה חשדה לעץ, נקבל ש־$H_1$ היה עץ שלם, כלומר מספר הצמתים בו קיים:

$$n-1=2^{h_1+1}-1=2^{h_2}-1$$

12) מקיים*
אם בהכנסת $m+1$ הוספנו רמה חשדה לעץ, נקבל ש־$H_1$ היה עץ שלם, כלומר מספר הצמתים בו קיים:

$$n-1=2^{h_1+1}-1=2^{h_2}-1$$

13) כיווני האי שוויון לא נכונים, ואז כשהופכים את כיווני האי שוויונים משהו משתבש בהוכחה. אם אתה מגדיל את האינדקס איך אתה מוכיח שה-Log ערך תחתון עדיין קטן מ-d? נראלי מוזר שלא השתמשת בתכונות הערך התחתון. צריך להגיד שבגלל שנשארת באותו גובה אז לא ייתכן כי log2 "קפץ" מעבר לערך השלם הבא, ולכן הערך התחתון נשאר אותו הדבר.

עבור צומת זה $x$, נקבל שהאינדקס שלו מקיים מהנחת האינדוקציה:

$$\lfloor \log_2(r(x)) \rfloor \geq d(r(x)) = h_1$$

ומכיוון שהאינדקס של האיבר שהוספנו, המקסימום, גדול לפי ההגדרה מהאינדקס של כל איבר אחר, וכן משום ששניהם באותה רמה, קרי אותו עומק, נקבל:

$$\lfloor \log_2(r(m+1)) \rfloor \geq \lfloor \log_2(r(x)) \rfloor \geq d(r(x)) = d(m+1)$$

14) בכללי כתבת הרבה פעמים "רמה", אבל זה לא נראלי מונח שהגדרנו. עדיף לשנות את זה לעומק כפי שדרשו בתרגיל.
15) יפה שהכנסת קצת אינפי חחחח, אבל המעבר הזה לא נכון:

$$\lim_{n\to\infty}\frac{\log^2(n)}{n}=\lim_{n\to\infty}\frac{2\cdot\log(n)\cdot\frac{1}{n}}{1}=\lim_{n\to\infty}\frac{2\cdot\frac{1}{n}}{1}=0$$

העלמת את הלוגוריתם. פשוט תדלג על השלב הזה ותגיד ישר שזה 0. זה ברור שפולינום מנצח לוגוריתם.
16) כדאי אולי לעבור לעבודה עם |s| ולא n. ובנוסף:

:$PopString()$

נמיר בלולאה את הרשימה $NextString$ למחרוזת בסיבוכיות $O(n)$ כאשר $n$ הוא מספר האיברים ברשימה, ולאחר מכן נכניס את המחרוזת ל־$Strings$.
לפי האלגוריתם שראינו בהרצאה, ההכנסה היא ב־$O(n)$ אורך המחרוזת.
אם במהלך ההכנסה הגענו לתו $\$$ קיים, כלומר המחרוזת קיימת כבר במבנה ־ ההכנסה

ההכנסה היא ב-O(n) **כאשר** n הוא אורך המחרוזת.

17) בעלים נאחסן את הערך 1? אין להם תת עץ..

שהכנסנו לעץ. נציין זאת רק לצורך השלמות, אך הדבר אינו רלוונטי לשאלה שכן לא נתבקשנו להרוס את המבנה).
בנוסף, נבצע סיור "$PostOrder$" על $LastSuffixTree$ לשמירת מידע נוסף:
עבור כל צומת, נשמור בו כמידע נוסף את מספר העלים בתת העץ שהוא שורשו.
כלומר עבור העלים (שבהם אנו עוברים בסיור לפני האבות שלהם) נאחסן את הערך 1.
כאשר אנו עולים לצומת כלשהו במהלך הסיור, נעבור על מערך האלף־בית $\Sigma$ השמור בצומת, ונסכום את המידע הנוסף שבכל אחד מהבנים המתאימים.
סיבוכיות האלגוריתם ליצירת עץ סיומות שראינו היא ליניארית באורך המחרוזת, וכן ראינו בהרצאה שהעץ המקבל הוא בעל מספר צמתים ליניארי באורך המחרוזת.

18) לא הבנתי את זה:

כאשר אנו עולים לצומת כלשהו במהלך הסיור, נעבור על מערך האלף־בית $\Sigma$ השמור בצומת, ונסכום את המידע הנוסף שבכל אחד מהבנים המתאימים.

מזתומרת נעבור על מערך האלף בית השמור בצומת ונסכום את המידע הנוסף? אתה מתכוון שאתה עכשיו שולף את המידע מכל אחד מהבנים?

19) המתקבל*

כאשר אנו עולים לצומת כלשהו במהלך הסיור, נעבור על מערך האלף־בית $\Sigma$ השמור בצומת, ונסכום את המידע הנוסף שבכל אחד מהבנים המתאימים.
סיבוכיות האלגוריתם ליצירת עץ סיומות שראינו היא ליניארית באורך המחרוזת, וכן ראינו בהרצאה שהעץ המקבל הוא בעל מספר צמתים ליניארי באורך המחרוזת.
לכן גם סיבוכיות הסיור היא ליניארית באורך המחרוזת, שכן עבור כל צומת אנו מבצעים מספר קבוע של פעולות השמה ב־$O(1)$ (משום שהאלף בית $\Sigma$ מגודל קבוע).

20) משפט לא כָל כָך ברור:

זאת משום שבעץ סיומות ראינו שכל מסלול מהשורש ועד לצומת כלשהו לתת מחרוזת שונה שמיוצגת על ידי שרשרות של ערכי הקשתות עד לאותו צומת.

אפשר למחוק את זה. המשפט שאמרת קודם ברור מספיק.

21) מה שרשמת פה לא בהכרח נכון:

בסך הכול, מספר המופעים של $r_2$ (ולכן של $r$) כתת מחרוזת שווה למספר העלים בתת העץ שמצאנו, מכיוון שכל עלה מציין סיומת שונה שמתחילה ב־$r_2$ ובפרט ב־$r$.
מכאן שהמידע הנוסף ששמרנו בצומת של $r_2$ בעת יצירת העץ נתון לקריאה ב־$O(1)$ ומכיל את הערך הרצוי, שנחזיר למשתמש.

מספר המופעים של r2 לא בהכרח שווה למספר העלים בתת העץ שמצאנו, כי r2 יכול להיות גם תת מחרוזת של סיומת נוספת. לדעתי כדאי אולי **להגיד שמספר המופעים של r כאשר r הוא תת מחרוזת של r2**, שווה למספר העלים ב...