הפיתרון למבחן מועד א'
בקורס "אלגוריתמים מבוזרים".

1) הפעלת אלג' Bellman-Ford.
בהתחלה ל-v יש משתנה $ctr(v)=0$ ולכל הצמתים האחרים $u \in V$, $v \neq u$, $ctr(u)=\infty$.
v שולח הודעה לכל שכניו עם $ctr(v)$.
כל צומת u שמקבל הודעה מ-w עם $ctr(w)$ משווה את $ctr(u)$ עם $ctr(w)+w(\{u,w\})$ ($w(\{u,w\})$ - משקל הצלע המקשרת בין u ו-w)
ואם $ctr(u)$ גדול ממש מ- $ctr(w)+w(\{u,w\})$ אז u קובע
$ctr(u) \leftarrow ctr(w)+w(\{u,w\})$.
כמו-כן אם u משנה את ערך $ctr(u)$ הוא גם מיידע הודעות לכל שכניו פרט ל-w עם עידכון ב-$ctr(u)$.
~~בסוף התהליך כל~~
כמו-כן u שומר את w (הצומת האחרון שגרם לעידכון $ctr(u)$) כהורו בעץ ה-SPT. הוא גם מודיע ל-w שהוא בנו, ורק w יכול לשמור את רשימת בניו.

__טענה-1:__ אחרי n סיבובים של ביחש, שבצומת u יקיים

$ctr(u) = dist_G(v,u)$.

כמו-כן הצומת w שאת צבעו u ~~[illegible]~~ ישמר בשלב הצבע כצבעת אביו בעץ T יקיים

$ctr(u) = ctr(w) + w(\{u,w\})$.

__הוכחה__

תהיה $P_{v,u}$ המסלה הקצרה ביותר (מבחינת משקל $w(P_{v,u})$) המקשרת בין v ל-u ~~[illegible]~~ בגרף G.

יהי ~~[illegible]~~ $|P_{v,u}|$ מס׳ הצלעות שהמסלה $P_{v,u}$ מכילה. קל לראות (באינדוקציה על $|P_{v,u}|$) שאחרי $|P_{v,u}|$ ~~[illegible]~~ סיבובים יתקיים

$ctr(u) \le w(P_{v,u}) = \sum_{e \in P_{v,u}} w(e)$.

כמו-כן קל לראות שלא ייתכן ש

$ctr(u) < w(P_{v,u})$,

כי אז צ״ל שקיימת מסלה אחרת $P'_{v,u}$ ~~שמסלה~~ שגם היא מקשרת בין v ל-u ומק״מת $ctr(u) = w(P'_{v,u}) < w(P_{v,u})$, וזאת סתירה לכך ש $P_{v,u}$ היא המסלה הקצרה ביותר המקשרת בין v ל-u ב-G.

כיוון ש $P_{u,v}$ הוא מסלול פשוט
(הסרת-מעגלים) $|P_{u,v}| \le n-1$ ומכאן הטענה
נובעת מיידית. החלק השני של הטענה
נובע ישירות מן החלק הראשון ומהתאור
האלג׳.

מ.ש.ל.

טענה 1 גוררת את נכונות האלג׳,
ואת החסם העליון של זמן הריצה
$Time(B-F) = O(n)$.

עבור סיבוכיות התקשורת נשים לב
~~של צומת u יכול לשדר את cost(u)~~
~~שלו לכל נותר~~ שבכל סיבוב ~~כל~~
~~בצומת~~ נשלחת הודעה אחת על כל
קשת של הגרף, ועל-כן בסה״כ נשלחות
$O(|E|\cdot n)$ הודעות במהלך האלג׳.

פיתרון שאלה 2)

ג. - נעשה פעם בהודעת broadcast
עם $ctr=0$. כל צומת v שמקבל את
ההודעה ~~הודעה w~~ מוסיף 1 ל-ctr, קובע את
$l(v)$ להיות שווה ל-ctr של ההודעה
שהוא קיבל ועוד 1, ושולח לכל בניו
הודעה עם ערך של ctr גדול ב-1
ממה שהיה בהודעה שהוא קיבל.
בתום התהליך הזה כל צומת v
יודע את ה-$l(v)$ שלו.

עתה נבצע pipelined convergecast של
מערכים
$n_0, n_1, \ldots, n_{Rad}$,
כאשר
$$n_i = |\{v \mid l(v) = i\}|$$
לכל $i \in [Rad]$.

במהלך ה-pipelined conv'st הזה כל
צומת v שולח לאביו בערך $Rad+1$ הודעות
$n_0(v), n_1(v), \ldots, n_{Rad}(v)$, כאשר
$$n_i(v) = |\{u \in T_v \mid l(u) = i\}|$$
! - T_v הוא תת-העץ של T שמושרש
ב-v.

כפי שלמדנו בכיתה, pipelined conv't
של m הודעות בעץ בעומק Rad
דורש. m+Rad סיבובים. כאן m = Rad,
ולכן זה לוקח בערך 2·Rad. עוד Rad
~~את~~ סיבובים נוספים נדרשו לחישוב
הערכים (ע"י ה- broadcast).
לכן $Time = O(Rad)$.
סיבוכיות התקשורת: על כל אחת מ-(n-1)
הקשתות של העץ נשלחו $O(Rad)$ הודעות.
על-כן $Comm = O(Rad \cdot n)$.

<u>הערה:</u> תלמידים רבים הציעו פיתרון
שבו כל צומת שולח מעלה פעם אחר
פעם את סכומי הערכים שהוא מקבל
מבניו. האלג' הזה נכון ועדיף על
זה שתואר לעיל, שכן הוא פשוט יותר
ויעיל להפעלה באותה מידה (אולי אף
יותר יעיל פי פקטור קבוע).

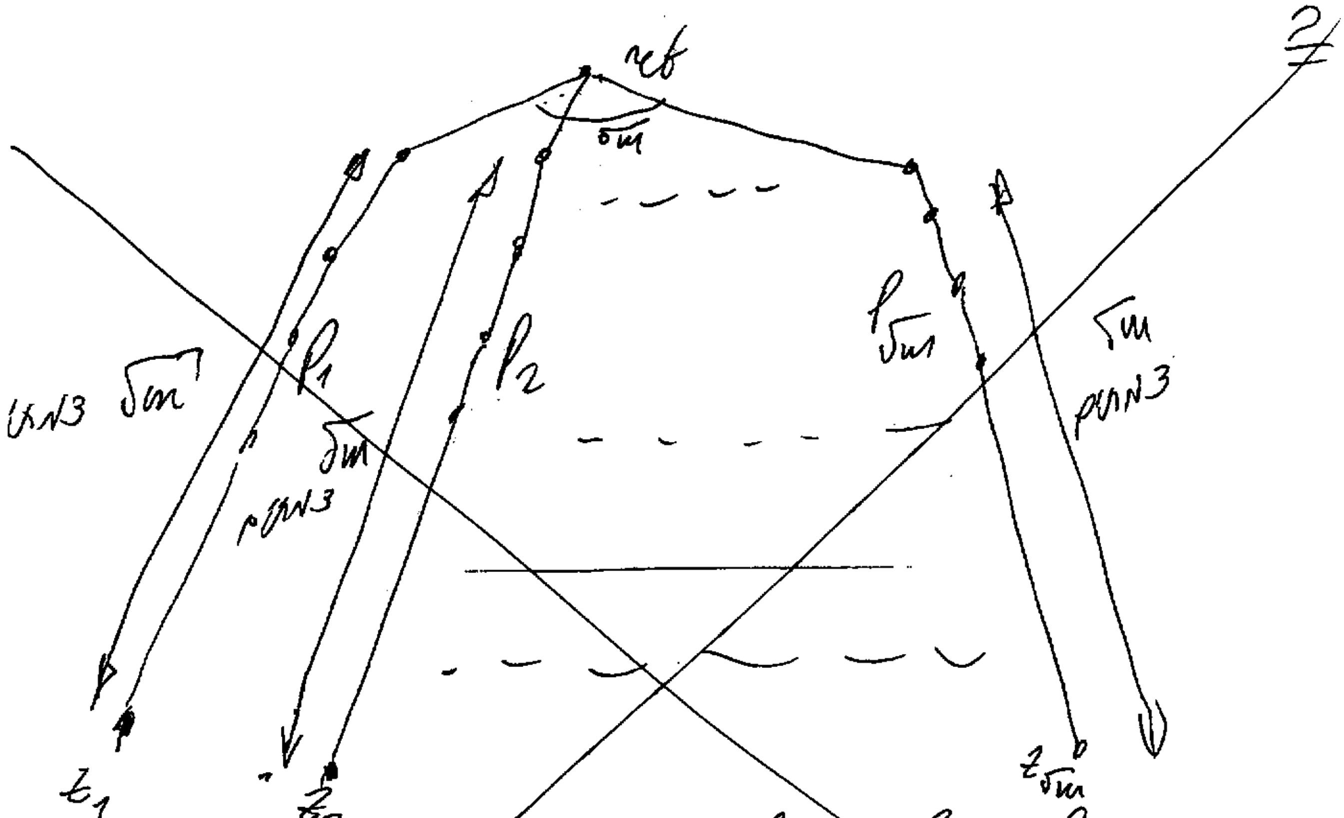
פיתרון שאלה 3)

6 נסתכל על אחד המספרים a
שהוא בין k הקטנים ביותר מתוך
m המספרים. נניח שבהתחלה הוא
נמצא בצומת v כלשהו ויהי
$P = (v = v_0, v_1, v_2, \dots, v_h = rt)$, $h \geq 0$
המסילה הפשוטה היחידה המקשרת בין v
ל-rt בעץ T.
לפי הגדרת ה-Rad, $h \leq Rad$.

~~נסתכל על צומת, v_i כלשהו, ו-$i \in [h]$~~
~~במסילה) P ונשאל כמה סבבים לכל היותר~~
~~עשוי המספר a להתעכב ב-v_i~~
~~הצומת~~

הצומת v_0 יודע איזושהי ת"ק S_0
של קב' m המספרים המקורית S. $(S_0 \subseteq S)$.
כיוון ש-a הוא בין k המספרים הקטנים
ב-S, אז הוא בין k " " "
בכל ת"ק של S ובפרט גם ב-S_0.
לכן תוך k סבבים לכל היותר
ישלח v_0 את a ל-v_1 שהוא אביו
בעץ (T, rt).
תהי S_1 קב' המספרים ש-v_1 יודע
ברגע שהוא קיבל את a מ-v_0 (כולל את a)

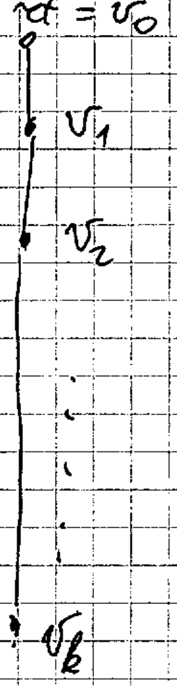
לפי אותו שיקול, תוך k סיבובים
לכל היותר יעביר v_1 את a ל-v_2,
[illegible]. כלה...
על-כן, תוך $Rad\cdot k \geq h\cdot k$
סיבובים יגע אלי את a.
כיוון שגם נבון לכ"א מ-k במספרים
כה קטנים כ-$\mathbb{Z}$, אנו מסיקים שתוך $Rad\cdot k$
סיבובים יגע אלי את B במספרים הנ"ל.
מ.ש.ל.



נסתכל על k על $\sqrt{m}$ כפל $m+1$ מחזורים
עבור m כאלו שמקיים ש-$\sqrt{m}$ הוא מספר
שלם. השורש k $\sqrt{m}$ הוא בוודאי אלי שערכנו
$\sqrt{m}$, ~~הוא~~ מחובר וכ"ו משפטן הוא קטן מן
מהלך ~~[illegible]~~ בה $\sqrt{m}$ 3מגיום.

3.ב. נסתכל על מסילה $v_0, v_1, \dots, v_k$

כך ש $v_0 = rt$.



הפיזור ההתחלתי של הפריטים
הוא ש v_k מחזיק בפריטים
$\{1, 2, \dots, k\}$,
v_{k-1} ב- $\{k+1, k+2, \dots, 2k\}$
$\vdots$
v_1 ב- $\{k(k-1)+1, \dots, k^2\}$.

נניח שבכל סיבוב הפריט
הגדול ביותר מבין k הקטנים בינהם
המוחזקים ע"י קודקוד נתון נשלח למעלה.

אזי אחרי k סיבובים הכל מוזז ב-1, דהיינו:
v_{k-1} מחזיק ב- $\{1, 2, \dots, k\}$
v_{k-2} מחזיק ב- $\{k+1, \dots, 2k\}$
$\vdots$
v_1 מחזיק ב- $\{k(k-2)+1, \dots, k^2-k\}$.

לכן השורש rt לומד את $\{1, 2, \dots, k\}$
רק אחרי כ- k^2 סיבובים.

פיתרון שאלה 4)

דוגמא. v שולח broadcast שיגרום convergecast שבמהלכו מחושבת הצלע ~~הקלה~~ הכבדה ביותר בטבעת. אז v שולח הודעה לקצוות ש הצלע הצעה שאומרת לכל שאר הצמתים שבה הטבעת ~~שהם~~ ~~זה~~ שסומנה להם שהוגדרה ב-MST, ~~של~~ ושאת הצלע בטבעת אין לבטל ב-MST (שכן לא שייכת ל-MST).
האלג' הזה דורש O(n) זמן ובנוסף נבצע מהצלע השינויים, דבר בגרף הזה יש רק ~~ב~~ מעגל סגור והצלע הכבדה ביותר נמחקת מן המעגל.
סיבוכיות התקשורת היא O(n) גם.

פיתרון שאלה 5

האלג׳ הוא גרסה של אלג׳ Pipelined MST שנלמד בכתה.

בשלב הראשון כל צומת לומד ~~את~~ את ה- Fragment-Ids של כל שכניו בגרף ומסמן כחולה כשמשלח קשת e={u,w}, ושולח גם Freq_Id(u) ו Freq_Id(w) (בנוסף ל- Id(u) ו- Id(w)).

כל סבב יקח זמן אחת.

עתה, בונים עץ BFS פורש של הגרף מושרש בצומת כלשהו (שנבחר באופן שרירותי - למשל, צומת בעל Id=1), ~~בעל ה Id הקטן ביותר~~ נקרא לעץ הזה T.

עתה מריצים את אלג׳ Pipelined MST על העץ T, אבל כל פעם ששולחים צלע, שולחים את ה-Fragment-Ids של קצותיה, וגם אף פעם לא שולחים צלע ששני הקצוות שלה, שייכים לאותו פרגמנט.

~~[illegible]~~

השורש אז מתחזקת רשימת L של
הצלעות שלא סוגרות מעגלים ביניהן, וכל צלע e
נוספת שהוא מקבל הוא בודק האם היא
לא סוגרת מעגל עם L ואם לא הוא מוסיף
את e ל-L.

כשהבנים של rt ב-T עוזבים,
ה-rt שולח את L ב-Downcast
לכל צמתי הגרף.

<u>טענה:</u> $|L| = O(\sqrt{n})$ עבור L שנשלח
ע"י ה-rt ב-Downcast.

<u>הוכחה</u>

מנכונות האלג'. Pipelined-MST נובע
שהצלעות של L הן בדיוק הצלעות
של MST שמוגדר בין הפרגמנטים.

כיוון שיש $O(\sqrt{n})$ פרגמנטים שמכסים
את כל הגרף, הרי שיש $O(\sqrt{n})$ צלעות
כאלה (כי הצלעות של MST שמוגדר
בין הפרגמנטים יוצרות עץ פורש של
גרף הפרגמנטים ובגרף זה יש $O(\sqrt{n})$ צמתים,
ולכן מספר הצלעות בעץ כזה הוא כמספר
הצמתים פחות 1)

לכן $|L| = O(\sqrt{n})$.

מ.ש.ל

מהטיעון שבהוכחת הטענה נובע
גם שתת-גרף שנבנה ע"י האלג'
הוא ה-MST כי הצלעות של L הן
בדיוק הצלעות המסרכות.

<u>טענה:</u> זמן הריצה של האלג' הוא $O(\sqrt{n})$

<u>הוכחה</u>
גודל [illegible]קב' הצלעות הבין-פרגמנטריות חסרת-מעגלים הגדולה
ביותר בגרף הוא $O(\sqrt{n})$.
לכן כל צומת צריך לשלוח לשורש
$O(\sqrt{n})$ או פחות צעדים.
לפי מה [illegible] כאן למעשה
Pipelined ~~Convegg~~Convergecast של $O(\sqrt{n})$ איברים בעץ
בעומק $O(\sqrt{n})$, וזה דורש $O(\sqrt{n})$ זמן
כפ' שלמדנו בכיתה.
(הטיעון המדויק הוא אנלוגי לגמרי
לטיעון של אלג' Pipelined MST שנעשה
~~בהרצ'~~ בכיתה.)

מ.ש.ל.

פיתרון שאלה 6.

מפעילים 7 פעמים את צעד
״רדוקציה במספר הצבעים״ שנלמד בכיתה.
הצעד הוא כזה: כל צומת v שאינו השורש
של העץ, משווה את color(v) (הצבע
של v בייצוג בינארי) עם color(parent(v))
(כאשר parent(v) הוא ההורה של v בעץ).
בהתחלה, לכל צומת v, color(v) = Id(v).
כעת מוצא את האינדקס i של הביט
כך ש $color(v)[i] \neq color(parent(v))[i]$
וקובע $color(v) \leftarrow bin\text{-}repr(i) \circ color(v)[i]$.
כפי שנלמד בכיתה, צעד כזה מוריד
את כמות הצבעים בצורה אקספוננציאלית
(אם לפני הצעד היו לנו צבעים בעלי
״גודל בינארי״ ~~צבעים~~ בן t ביטים, הרי
שאחרי הצעד יהיו צבעים בעלי ״גודל
בינארי״ של $\lceil \log t \rceil + 1$ ביטים),
והוא שומר על תקינות הצביעה.
צעד זה אפשר לעשות מקומית
(בכל צומת בהינתן הצבע של ההורה) ללא
צורך בתקשורת.

אחרי הצעד הבא כל מעבד מודיע לשכן שלו
את כל. ~~[illegible]~~ צבעו
מהו. הצבע בקוד שלו. גם דורש ~~[illegible]~~ סיבוב
אחד.

ומפעילים את שני הצעדים הנ"ל בלולאה
שבכל פעם ~~[illegible]~~ אחד אחרי השני,
כמות הצבעים שמתקבלת היא קטנה או
שווה מ-

$O(\log^{*} n)$.

ברור כי מתקבלים ~~$O(\log n)$~~ צבעים,
וכל הפעלה של הצעד מורידה את כמות
הצבעים אקספוננציאלית
(בלי תקשורת בכלל אפשר להשיג
$O(\log n)$ צבעים, כי מראש כל צומת
יודע את ה-id שלו ואת ה-id של שכנו;
תוך סיבוב אחד אפשר להוריד את
כמות הצבעים ל- $O(\log\log n)$, וכך הלאה...)