תרגיל 2- אלגוריתמים 67504:

שמות: דן קורנפלד, מתן קמינסקי

שאלה 1:

1. כדי להוכיח את חוקיות האלגוריתם, נצטרך להוכיח שהפלט הוא עץ פורש (כלומר גרף קשיר, מכיל את כל קודקודי G, וחסר מעגלים).
2. ראשית, האלגוריתם מכניס צלעות לגרף T אך ורק אם הצלע לא סוגרת מעגל, כלומר בכל שלב אנחנו נמצאים ב1 מתוך 2 המצבים הבאים:
   (a) גרף עם יותר מרכיב קשירות יחיד.
   (b) גרף עם רכיב קשירות יחיד, ללא מעגלים (עץ).

★ אם האלגוריתם T קשיר, סיימנו. נניח בשלילה כי האלגוריתם אינו מחזיר גרף קשיר. אזי, קיימים לפחות 2 רכיבי קשירות. נסמנם T_1, T_2 בהתאמה, מאחר והם רכיבי קשירות שונים, לא קיים מסלול בין אף קודקוד ב-T_1 לקודקוד ב-T_2 ולהפך.
נגדיר קבוצה E, אשר מכילה את כל הצלעות בגרף G (גרף הקלט) שמקשרות בין 2 רכיבי הקשירות.
הקבוצה אינה ריקה, שכן אם הייתה, G היה לא קשיר, בסתירה לנתון.
הוספת צלע מ$E \rightarrow T$ אינה יכולה לסגור מעגל, שכן אם הייתה, היינו מסיקים שקיים מסלול בין קודקודים מרכיב אחד לשני, בסתירה להנחה. כלומר, הוספת צלע יחידה אינה יוצרת מעגל.
האלגוריתם עובר על כל הצלעות, קיימת צלע כלשהו ב-E שלא סוגרת מעגל, ולכן בהכרח האלגוריתם יוסיף אותה ל-T, כך נקבל גרף קשיר, בסתירה להנחה שיש לפחות 2 רכיבי קשירות.

★ חסר מעגלים: מאופן פעולת האלגוריתם, בכל מצב בו צלע יכולה לסגור מעגל, האלגוריתם לא יוסיף אותה, ולכן לא קיימים בגרף מעגלים.

שאלה 2:

1. נציע את האלגוריתם הבא ונוכיח את נכונותו.
2. אלגוריתם:
   (a) נאתחל קבוצה ריקה B, שתכיל את הפתרון החוקי: $y_1,..,y_m$
   (b) נכניס ל-B, את x_1
   (c) נעבור על כל הx_i לפי הסדר, מ-x_1
   (d) נבדוק עבור כל נקודה, האם הקטע האחרון שהאלגוריתם הוסיף מכיל את הנקודה, כלומר: $x_i \in [y_j, y_{j+1}]$, אם הקטע מכיל את הנקודה הנוכחית, נמשיך לנקודה הבאה, אם לא אז נבצע את (e).
   (e) בכל פעם שקיימת נקודה ללא קטע, נוסיף קטע חדש שמתחיל באותה נקודה, ונוסיף ל-B
   (f) נרוץ עד סיום כל הנקודות
3. חוקיות:
   (a) נניח בשלילה כי האלגוריתם החזיר את קבוצה B, כך שקיים x_i שאינו מוכל באף אחד מהקטעים.
   (b) משמעות הדבר היא: שלא קיים $j \in [n]$, שעבורו $x_i \in [y_j, y_j + 1]$.
   (c) מאופן פעולת האלגוריתם, ברגע שהאלגוריתם מגיע לנקודה כלשהי, והנקודה אינה מוכלת בקטע האחרון שהתווסף, האלגוריתם יוסיף קטע חדש שיכיל **ויתחיל** ב- x_i, על כן, נקבל סתירה לפעולת האלגוריתם.
4. מינימליות- אופטימליות:
   (a) נסמן $B = \{y_1,..,y_m\}$ את הפתרון החמדן. בשביל להראות את האופטימליות נוכיח את הטענה הבאה:
   טענת האינדוקציה: לכל $k \in \{1,..,m\}$ קיים פתרון אופטימלי מהצורה $C = \left\{y_1,...,y_k, c_{k+1},...,c_{m'}\right\}$.
      i. <u>בסיס</u>: עבור $k = 1$, מחוקיות האלגוריתם מתקיים ש$y_1 = x_1$, מכאן כל פתרון חוקי מתחיל בx_1, בהכרח כל פתרון אופטימלי מתחיל גם כן ב-$y_1 = x_1$.
      ii. <u>הנחת האינדוקציה</u>: נניח את הנכונות עבור $k - 1$, כלומר קיים פתרון אופטימלי מהצורה: $C = \left\{y_1,...,y_{k-1}, c_k,...,c_{m'}\right\}$, כעת נוכיח עבור k.
      iii. <u>צעד</u>: אם הקטע $x_n \in [y_k, y_{k+1}]$, כלומר קיים קטע המכיל את הנקודה האחרונה (ואלה שקדמו לה), לכן סיימנו.
      עתה, נסתכל על מצב הביניים שבו $k < m'$.
      נחליף את הקטעים y_k, c_k ונראה שהחוקיות והאופטימליות נשארת.
         A. החלפה זו לא פוגעת בכל ה-x-ים שלפני x_k, היות ולא הסרנו קטעים בתחום הרלוונטי.
         B. נניח בשלילה ש$y_k < c_k$. לפי האלגוריתם שהצענו, נרצה להוסיף את y_k כאשר לא קיים קטע המכיל את הנקודה x_k, בסתירה להיות C פתרון חוקי ואופטימלי.
         C. לכן, $y_k \geqslant c_k$, אם $y_k = c_k$ אין שום שוני בקבוצות עד האיבר ה-k, ולכן הוכחנו, נוכיח עבור $y_k > c_k$.

D. במצב בו $y_k > c_k$, קיימת נקודה כלשהי $x_p = c_k$. $x_p \in [y_{k-1}, y_{k-1}+1]$, היות והאלגוריתם שהצענו לא הוסיף קטע חדש המתחיל בנקודה הזו, ומכיוון שהאלגוריתם שהצענו והאלגוריתם האופטימלי מסכימים על כל הצעדים עד $k-1$ כולל, הפתרון האופטימלי כולל את הקטע $[y_{k-1}, y_{k-1}+1]$, ובנוסף את הקטע המתחיל ב- x_p.

E. לפי האלגוריתם, כל הנקודות בקטע $[x_p, x_k] \subseteq [y_{k-1}, y_{k-1}+1]$, מאופן פעולת האלגוריתם אשר עובר על כל הנקודות מההתחלה עד הסוף ומוסיף קטע רק כאשר הנקודה לא מוכלת בקטע האחרון.

F. מכיוון שיש חפיפה בין 2 הקטעים, נשתמש בקטע המתחיל ב x_p ונזיז אותו כדי שיתחיל בy_k. עתה, כל הנקודות עד $x_k + 1$ מוכלות בקטעים, לא פגענו בכל הקטעים החל מ-c_{k+1}, לכן השארנו פתרון חוקי.

iv. לכן, הוכחנו את הצעד, כלומר האלגוריתם מסכים עד הצעד ה-k ומכאן הפתרון החמדני הוא פתרון חוקי.

5. זמן ריצה:

(a) זמן הריצה של האלגוריתם הוא $O(n)$ מכיוון שאנחנו רצים פעם אחת על כל x

שאלה 3:

1. נציג אלגוריתם לפתרון הבעיה.
2. אלגוריתם:
   (a) נגדיר $L = \varnothing$ אשר ישמור את כמות הליטרים שהאלגוריתם הכניס בכל תחנת דלק.
   (b) עבור כל תחנה a_i, $1 \leqslant i \leqslant n-1$, נבדוק מה המרחק בין a_i עד a_{i+1}, ונוסיף למיכל בדיוק את המרחק בליטרים של דלק, שכן הרכב יגיע עם 0 ליטרים לתחנה הבאה.
   ונוסיף ל- L את הכמות שהוספנו בתור $l_i = |a_{i+1} - a_i|$
      i. הערה: נתון שהמרחק בין כל 2 תחנות הוא קטן או שווה ל-N, ולכן בהכרח נוכל למלא כמות ליטרים שאינה תחרוג מגודל המיכל.
   (c) מכאן הרכב יגיע לתחנה ה-a_n עם 0 ליטרים, ונמלא את הכמות המינימלית ההכרחית (הוכחה בהמשך)
3. חוקיות:
   (a) מאופן פעולת האלגוריתם אנו מחזירים קבוצה בגודל n שמבטאת את כמות הליטרים שהוספנו בכל תחנה, ומכאן חוקיות הפתרון.
4. אופטימליות:
   (a) נוכיח באינדוקציה את האופטימליות.
   (b) טענת האינדוקציה: לכל $1 \leqslant k \leqslant m$ קיים פתרון אופטימלי C שמסכים עם הפתרון L על k האיברים הראשונים של L.
   (c) בסיס: $k = 1$, l_1 היא כמות הליטרים המינימלית כדי לנסוע בין התחנה הראשונה, לתחנה השניה. לא נוכל להתקדם עם כמות ליטרים קטנה מl_1, לכן הבסיס תקין
   (d) נניח כי קיים פתרון אופטימלי $C = \{l_1,.., l_{k-1}, c_k,..., c_n\}$, נבנה קבוצה $C' = \{l_1,.., l_k, c_{k+1},..., c_n\}$, נראה כי C' הוא פתרון חוקי ואופטימלי.
   (e) נגדיר j כך ש- $k+1 \leqslant j \leqslant n$, כאשר j הוא האינדקס המינימלי שעבורו $c_j > 0$.
   (f) נשנה ב-C' את ה-c_{k+1} להיות $c'_{k+1} = |a_j - a_{k+1}| + \varepsilon$, כאשר $\varepsilon \geqslant 0$ מבטא את כמות הליטרים איתם הגיע הרכב לתחנה a_j ב-C המקורי אליהם נוסיף את c_j וכן הלאה, ונמשיך לפי C המקורי
   (g) הערה: כמות הדלק $|a_j - a_{k+1}| + \varepsilon$ בהכרח קטנה מ-N היות ולפי הפתרון המקורי, $a_k,.., a_{j-1}$ לא תדלקנו, לכן בהכרח המרחק קטן מ-N.
   (h) חוקי: בכל התחנות מ-$l_1 \to l_k$ יש מספיק דלק למעבר בין תחנות (לפי האלגוריתם שבנינו, ותקינות הקלט), בנוסף, מ$c_j \to c_n$ כמות הדלק לא השתנתה ומחוקיות פתרון C, אנו עומדים בדרישות. כמו כן, עבור התחנות $a_{k+1} \to a_j$, לפי המפורט לעיל יש מספיק דלק לנסוע.
   (i) סכום הקבוצות: בחלק הראשון: $l_1 \to l_{k-1}$ המרחקים בין הפתרון L ל-C (האופטימלי) זהה, באותו אופן $c_j \to c_n$ הסכום זהה גם כן,

בפתרון האופטימלי C, האלגוריתם מגיע לתחנה a_k עם 0 ליטרים של דלק, ומאחר וכל הקטעים בין $c_{k+1} \to c_{j-1}$ שווים ל-0, נסיק כי בתחנה a_k האלגוריתם מילא כמות דלק שמספיקה להגיע ל-a_j עם כמות דלק ε שנשארה (לפי C המקורי). כלומר $c_k = |a_j - a_k| + \varepsilon$

$$c_k = |a_j - a_k| + \varepsilon = |a_j - a_{k+1} + a_{k+1} - a_k| + \varepsilon = \underbrace{\varepsilon + |a_j - a_{k+1}|}_{c'_{k+1}} + \underbrace{|a_{k+1} - a_k|}_{l_k} \Longrightarrow$$

$$\Longrightarrow c_k = c'_{k+1} + l_k$$

כמו כן $c_k + \underbrace{c_{k+1}}_{=0} = l_k + c'_{k+1}$, וכן כל האיברים $c_{k+2} \to c_{j-1}$ שווים ל-0 ב-2 הקבוצות, ולכן הסכום שווה, ומכאן האופטימליות של C'

(j) מהאינדוקציה נסיק את האופטימליות של L עבור $k = n$, ולכן L אופטימלי ■

שאלה 4:

סעיף 1:

1. האלגוריתם לא אופטימלי, הפרכה: המספר 12: אם עובדים לפי האלגוריתם נקבל:
$12 \to 11 \to 10 \to 9 \to 8 \to 4 \to 2 \to 1$, כלומר 7 צעדים
ואילו ניתן להגיע בעזרת 2 הפעולות הנתונות בעזרת 4 צעדים: $12 \to 6 \to 3 \to 2 \to 1$.

סעיף 2:

1. נציע את האלגוריתם הבא:
   (a) נגדיר רשימה ריקה=L
   (b) נעבור בלולאה כל עוד המספר גדול מ-1
   (c) בכל איטרציה, נבדוק האם המספר הוא זוגי, או אי-זוגי, אם המספר אי זוגי: נוריד את האיבר ב-1, ונוסיף לרשימה L, את הערך $increase$,
   עבור מספר זוגי, נחלק ב-2, ונוסיף לרשימה $double$.
   (d) כשנגיע ל-1 נעצור, ונהפוך את הרשימה ($reverse$) ונחזיר את האורך שלה.
2. חוקיות: היות ואנחנו ממשיכים בתהליך כל עוד המספר חיובי וגדול מ-1, ואנחנו מבצעים פעולת חיסור עבור מספר אי זוגי (והתוצאה נשארת טבעית), כמו כן עבור מספר זוגי מבצעים חילוק (והתוצאה נשארת טבעית גם כן), אז היות ו-k יורד בצורה מונוטונית, בהכרח נגיע ל-1.
3. אופטימליות:
   (a) נוכיח את האלגוריתם באינדוקציה, כך שהאופטימליות נמדדת בכמות הפעולות שיש לבצע.
   (b) בסיס: $k = 0$, לא נבצע דבר, ישר נחזיר רשימה ריקה (לא צריך לבצע כלום)
   (c) הנחה: נניח כי קיים פתרון אופטימלי $C = \{l_1,.., l_{k-1}, c_k,..., c_n\}$, נבנה קבוצה $C' = \{l_1,.., l_k, c_{k+1},..., c_n\}$, נראה כי C' הוא פתרון חוקי ואופטימלי.
   (d) נחלק את הצעד ה-l_k ל-2: אם יש שוויון בין הפעולות, או לא
      i. אם יש שיוויון, מעולה, נשאיר את הפעולה ה$c_k = l_k$
      ii. אם יש שוני בין הפעולות:
         A. אם האלגוריתם האופטימלי מבצע חילוק, כלומר המספר זוגי, בהכרח גם האלגוריתם שהצענו מבצע חילוק
         B. אם האלגוריתם האופטימלי מבצע חיסור, ואנחנו מבצעים חילוק יש 2 אפשרויות:
            ★ אם הגענו ל$k = 2$, והאלגוריתם מבצע חיסור ואנחנו חילוק, ב-2 המקרים סיימנו כי הגענו ל$k = 1$
            ★ אחרת, אם ביצענו חילוק, המספר בהכרח זוגי, ובהכרח האלגוריתם האופטימלי יצטרך לאחר חיסור לבצע חיסור נוסף, שכן הוא לא יכול לחלק במספר אי זוגי. כל זוג פעולות חיסור יכולות להתבצע על ידי פעולת חיסור אחת במידה וקיימת לפניה פעולת חילוק, לכן אם האלגוריתם האופטימלי המוצע מבצע פעולת חיסור במקום חילוק, בהכרח האלגוריתם לא אופטימלי מההסבר לעיל.
            ★ אם אין עוד חלוקות ב-2 באלגוריתם האופטימלי, היה ניתן לצמצם בחצי את כמות פעולות החיסור הנוספות על ידי פעולת חילוק יחידה ולאחריה חיסור אחד בסתירה לאופטימליות
   (e) לכן, לאור המפורט לעיל בהכרח חוקי, ויש פתרון אופטימלי כלשהו, הפתרון החמדן (המפורט) הוא גם פתרון אופטימלי ■

שאלה 5:

1. נציע את האלגוריתם הבא: נמיין את הרשימות A, B מהגדול לקטן, ונחזיר רשימה $C = [c_1, ..., c_m]\ \forall c_i = (a_i, b_i)$.
2. זמן ריצה: $o(mlog(m))$, היות ואנחנו ממיינים כל רשימה, ולאחר מכן עוברים על כל האיברים ברשימה הממוינת כדי ליצור את C
3. חוקיות: הפתרון חוקי היות ואנחנו משתמשים בכל איבר $a \in A, b \in B$ פעם אחת, ומחזירים קומבינציה כלשהי של צירופי המספרים.
4. הסבר מתמטי עבור אופטימליות המקסום: בפירוט זה, הרשימות A,B ממוינות מהקטן לגדול
   (a) $n = 2$: $a_1^{b_2} \cdot a_2^{b_1} = a_1^{b_1} \cdot a_1^{b_2 - b_1} a_2^{b_1} \leqslant a_1^{b_1} \cdot a_2^{b_2 - b_1} a_2^{b_1} = a_1^{b_1} \cdot a_2^{b_2}$ $(a_1 \leqslant a_2, b_1 \leqslant b_2)$
   (b) צעד: נניח כי עבור $2, .., k-1$ כל קומבינציה של מכפלת $a_i^{b_j}$ קטנה או שווה ממכפלת $a_i^{b_i}$.
   נראה נכונות עבור k
   (c) נחלק למקרים:
      i. אם האיבר ה-k הוא $a_k^{b_k}$ אזי מהנחת האינדוקציה מכפלת $k-1$ האיברים הראשונים קטנים או שווים ממכפלת $a_i^{b_i}$, וקיבלנו את הנדרש.
      ii. אחרת, יהא $1 \leqslant j \leqslant k$ כך שעבורו $a_k^{b_j}$, מהנחת האינדוקציה השלמה מתקיים שהמכפלה $a_k^{b_j} \cdot a_z^{b_k} \leqslant a_k^{b_k} \cdot a_z^{b_j}$, $a_k \geqslant a_z$, $b_k \geqslant b_j$ (מקרה הבסיס)
      iii. לכן, המכפלה של כל k האיברים קטנה שווה למכפלה של כל ה-k האיברים לאחר השינוי שהצגנו לעיל, כלומר $a_k^{b_k}$ כפול מכפלה של $k-1$ איברים, אולם מהנחת האינדוקציה המכפלה הנל קטנה מהמכפלה $\prod_{i=1}^{k} a_i^{b_i}$ כנדרש ■