תרגיל 3- אלגוריתמים 67504:

שמות: דן קורנפלד, מתן קמינסקי

שאלה 1:

1. נסמן: את הקבוצה S להיות קבוצת צלעות מגרף G שאינן סוגרות מעגל.
2. כעת, נגדיר את הקבוצה S' להיות מהצורה הבאה: $\forall e \in S$ נתאים וקטור $v_e \in \mathbb{Z}_2^n$ (כלומר וקטור בגודל n- כמימד הקודקודים, המכיל קורדינטות $1 \setminus 0$ בלבד), שערכו 1 בקורדינטות המתאימות לקודקודי הצלע e בגרף ו-0 בשאר הקורדינטות.
3. ★ - נשים לב מאחר והוקטורים ב$\mathbb{Z}_2$ מתקיים ש:$1 + 1 = 0$וכי לא קיים ערך אחר במרחב, ומלבד $1 \setminus 0$.
4. נניח בשלילה כי קבוצה S' תלויה לינארית לכן, ניתן להסיק כי קיימת תת קבוצה A כך שחיבור כל הוקטורים בקבוצה יצא וקטור ה-0, ומקדם כל וקטורים בצירוף הלינארי יהיה 1. מכאן, נסיק שלכל קורדינטה חייבת להיות מספר זוגי של וקטורים שערכם 1 בקורדינטה, כדי שהקורדינטה תתאפס.
5. נוכיח באינדוקציה את הטענה לפניה: בקבוצה A קיימים k וקטורים בהם כל קורדינטה מופיעה פעמיים למעט 2 קורדינטות.
6. <u>בסיס:</u> יהא וקטור $e \in S'$ נסמן את הקורדינטות: $\begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix}$ כך ש$e_i, e_j = 1$ וכל שאר הקורדינטות שוות ל-0.
   מההנחה לעיל קיים וקטור נוסף שעבורו $e_i = 1$, וגם $e_j = 0$ (שכן אחרת זה וקטור e).
   לוקטור הנוסף קיים $e_k = 1$ כנדרש ($k \neq j$)
7. הנחה: נניח שהטענה מתקיימת עבור $k - 1$ וקטורים ונוכיח עבור k.
   כלומר: קיימים $k - 1$ וקטורים שעבורם למעט זוג קורדינטות, כל קורדינטה מופיעה בדיוק פעמיים.
   מההנחה לעיל בקבוצה A כל קורדינטה מופיעה כמות זוגית של פעמים, לכן בהכרח חייב להיות וקטור נוסף בקבוצה A שערכו באחת הקורדינטות שמופיעות פעם אחת הוא 1.
8. נחלק למקרים:
   (a) מקרה 1: הוקטור שהוספנו עתה, בעל קורדינטות שתואמות ל-2 הקורדינטות שמופיעות פעם 1, למעשה סכום של כל הוקטורים שהוספנו הוא וקטור ה-0, כלומר הוספנו את כל קבוצה A
   (b) מקרה 2: הוקטור שמוסיפים גורם לכך שיש לנו קורדינטה אחת תואמת, וקורדינטה חדשה שנפתחת.
   (c) מקרה 3: וקטור שמוסיפים גורם לכך שיש לנו קורדינטה אחת תואמת, והקורדינטה השניה גורמת לסכום של 3 (כלומר היה אפס, ועכשיו הסכום הוא 1). אופן בחירת הוקטור היא בצורה המשכית: הוקטור הבא נבחר לפי הקורדינטה החופשית שנשארה והוקטור ה-$k - 1$, כלומר אם נחזור בבחירת הוקטורים, בהכרח נגיע לוקטור שבו הקורדינטה שמופיעה 3 פעמים נמצאת שם ושסכום והוקטורים שעברנו כדי להגיע לוקטור זה, יהיה 0.
9. לכן, מהוכחת האינדוקציה קיימת בהכרח קבוצה שאם נוסיף לה וקטור נקבל שכל קורדינטה שערכה 1 מבין הוקטורים בהכרח אומרת שקיימים 2 וקטורים בדיוק המכילים את הקורדינטה הנ"ל.
10. מאחר וכל קורדינטה מייצגת קודקוד, ומאופן בחירת הוקטורים באינדוקציה ניתן לראות שאם נתחיל מקודקוד מסוים בוקטורים שבחרנו, נוכל לחזור בחזרה לאותו קודקוד, כלומר סגרנו מעגל, בסתירה

לכך שאין מעגלים.

כיוון שני: אם קבוצת וקטורים היא בת"ל אז לא קיים מעגלים:

1. נראה את ההוכחה עבור טענה שקולה: אם קיים מעגל, אזי קבוצת הוקטורים היא ת"ל
2. תהא A קבוצה של וקטורים בת"ל ונניח בשלילה כי קבוצת הצלעות המתאימה סוגרת מעגל.
3. מההסבר לעיל פירטנו שעבור מעגל, כל קודקוד צריך להיות בדיוק פעמיים בהצגת הצלעות, כלומר קיימת ב-A תת קבוצה שמכילה בדיוק את צלעות המעגל.
4. לפי (3) ולפי אופן בניית הוקטורים נסיק כי כל קורדינטה שמבטאת קודקוד במעגל צריכה להופיע בדיוק ב-2 וקטורים.
5. לכן, כשנסכום את כל הוקטורים בתת הקבוצה, למעשה נסכום כל קורדינטה פעמיים ונקבל את וקטור ה-0, כלומר הקבוצה היא ת"ל, בסתירה להנחה ■

שאלה 2:

סעיף 1:

1. <u>נגדיר את תת הבעיות</u>: מציאת הסדרות העולות שאורכן מקסימלי עבור תת הסדרה $(a_1,....,a_n)$ לכל $1 \leqslant i \leqslant n$.
2. נוסחת הרקורסיה ופירוט: נגדיר את נוסחת הרקורסיה הבאה: נסמן את $T[i]$, אורך הסדרה העולה עם האורך המקסימלי. $$T[i] = \begin{cases} max\left\{\{T[j] : i > j, a_i > a_j\}, \{0 : i > j, a_i \leqslant a_j\}\right\} + 1 & i > 1 \\ 1 & i = 1 \end{cases}$$
   נוסחת הרקורסיה עבור תת הקבוצה באורך 1 יהיה 1 (תנאי בסיס)
   לכל תת סדרה באורך גדול יותר, נבחר את המקסימום: אם קיימת תת סדרה עולה וניתן להמשיך אותה ו-0 עבור מצב בו כל האיברים הבאים לפני $a[i]$ גדולים ממנו, לזה נוסיף 1, כלומר האיבר ה-i.
3. **<u>תיאור הטבלה, סדר המילוי ואופן חילוץ הפתרון</u>:**
   <u>גודל הטבלה</u>: $1 \times n$: תא עבור כל $T[i]$ המכיל את האורך המקסימלי של הסדרה העולה.
   <u>סדר המילוי</u>: נתחיל מהאינדקס 1 ונעלה מעלה לפי הסדר עד שנגיע לסוף הסדרה, כלומר האינדקס n.
   <u>אופן חילוץ הפתרון</u>: נעבור על כל הטבלה מהאינדקס ה-1 עד האינדקס ה-n, ונחזיר את הערך המקסימלי.
4. ניתוח זמן ריצה: $O\left(n^2\right)$ סכום סדרה חשבונית $\sum_{i=1}^{n} i = \frac{n(n-1)}{2} = O\left(n^2\right)$

סעיף 2:

1. במידה והיו מבקשים את הסדרה עצמה בנוסף לאורכה, היינו שומרים שורה שלמה נוספת בטבלה, כך שכל עמודה מתאימה ל-i המתאים, ונשמור שם את הבחירה שעשינו עבור אותה איטרציה, כלומר אם בחרנו ב$T[i]$ לשמור בבחירה את $T[j] + 1$, היינו שומרים בתא את j, אם היינו בוחרים את 0, היינו מסמנים את i בתא הנוסף.

סעיף 3:

נוכיח את נכונות הפתרון (חוקיות ואופטימליות)

1. חוקיות: בסיום האלגוריתם מקבלים אורך תת סדרה עולה.
2. אופטימליות: נוכיח באינדוקציה שלכל i האלגוריתם מייצג את אורך תת הסדרה הגדול ביותר מ-1 עד i כך ש $a[i]$ הוא האיבר האחרון בתת הסדרה.
   (a) בסיס: $i = 1$, נחזיר את $a[1]$
   (b) הנחה: נניח כי הטענה נכונה לכל $i - 1$ ונוכיח עבור i.
   (c) צעד:
      i. כל האיברים לפני האיבר ה-i גדולים ממנו, לכן לפי הגדרת הקבוצה אורך הסדרה יהיה 1
      ii. קיימת תת סדרה עולה בתת הסדרה $(a_1,....,a_{i-1})$, לפי אופן מילוי מקום ה-$T[i]$ האלגוריתם מסתכל על כל המקומות בטבלה שקטנים מ-i, נסתכל על המקסימום מבין האורכים וגם ש$j < i$, $a[j] < a[i]$, שהאורך מקסימלי, ונוסיף לו 1, (עבור $a[i]$), מכיוון שמהנחת האינדוקציה $T[j]$ הוא האורך המקסימלי שעבורו $a[j]$ הוא האיבר האחרון בתת הסדרה ומהיותה תת הסדרה הארוכה ביותר, נסיק כי $T[i]$ יכיל את אורך תת הסדרה המקסימלית המכילה את $a[i]$ כאיבר האחרון.

(d) לכן, האלגוריתם יעבור בסוף על הטבלה המכילה את אורך הסדרות הארוכות ביותר המסתיימות ב$a[i]$ ויחזיר את האורך של תת הסדרה הארוכה ביותר, כמבוקש■

שאלה 3:

נתאר אלגוריתם לפתרון:

1. ראשית, אם קיבלנו רצף ספרות בגודל 0 נחזיר אפס, אחרת:
2. נעבור על הספרות מהשמאלית ביותר לימנית ביותר ונחפש מופעים של הספרה 0.
3. במידה ויש מופע המכיל את הספרה 0, אם לא מתקיים שקיימת ספרה (2\1) משמאל ל-0, לא ייתכן שקיים פתרון חוקי, ונחזיר 0.
4. שנית, נגדיר טבלה בגודל $1 \times n$ ריקה, כך שכל תא יכיל את כמות האופציות לפענוח הקידוד לקטעי ספרות מהשמאלית ביותר עד לתא ה-i.
5. נתחיל מהספרה השמאלית ביותר ונשים בתא הראשון את הערך 1.
6. לכל הספרה ה-i, נבדוק את כמות האופציות עבור הגודל $i-1$ ונבדוק האם צירוף הספרות: $i-1, i$ הוא אופציה תקינה (1-26), במידה וכן: נוסיף לסכום שהגדרנו את כמות האופציות עבור מילה בגודל $i-2$, ונמקם בטבלה את הסכום שלהם.
7. לבסוף, נחזיר את התא ה-n מהטבלה.

הערה: ניתן להוכיח כי רצף הספרות הינו חוקי ותקין היות והאלגוריתם בודק את תקינות הקלט בתחילתו.

הערה: נגדיר כל ספרה כ-a_i עבור הספרה ה-i בקטע

1. <u>ניסוח בהיר של תתי הבעיות</u>: נסתכל על רצפי הספרות מהספרה הראשונה (השמאלית ביותר) עד הספרה ה-i, לכל $1 \leqslant i \leqslant n$, כמספר המקרים לפענוח הקידוד.
2. <u>נוסחת רקורסיה והסבר לבניתה:</u>

נוסחת הרקורסיה:

$$T[i] = \begin{cases} 1 & i = 1 \\ T[1] + 1 \;\; if : a_1 a_2 \in [26] \setminus \{10, 20\} & i = 2 \\ 0 : \qquad elif \;\;\; a_1 a_2 \in \{10, 20\} & \\ T[1] \qquad else & \\ T[i-1] + T[i-2] \; if \; (a_{i-1} \neq 0 \wedge a_{i-1} a_i \in [26] \setminus \{10, 20\}) & i > 2 \\ T[i-2] \; if \; a_{i-1} a_i \in \{10, 20\} & \\ T[i-1] \; else & \end{cases}$$

הסבר: עבור רצף באורך 1 קיימת אופציה אחת, עבור רצף באורך 2, נבדוק אם הצירוף מאפשר אות ב $A - Z$, אם כן נוסיף אופציה נוספת אחרת לא

לכל מקרה גדול יותר: נעבוד בצורה דומה: אם שילוב 2 הספרות האחרונות מהוות אות כלשהי, נסכום את כמות האופציות בגודל $i-2$ ו $i-1$, אחרת נשאיר את הסכום כסה"כ האופציות בגודל $i-1$.

כמו כן, הוספנו את המצבים בהם האיבר שאנחנו מסתכים עליו הוא 0, במצב כזה לא נוסיף איברים נוספים היות וה-0 חייב להצטרף לספרה אחרת.

3. <u>תיאור של הטבלה, סדר המילוי שלה ואופן חילוץ הפתרון:</u>

כאמור, נגדיר טבלה בגודל $1 \times n$ כך שכל תא ה-i יכיל את מספר האופציות לצירוף חוקי עבור תת רצף המתחיל בספרה השמאלית עד הספרה ה-i.

סדר המילוי: כמפורט באלגוריתם נמלא את הטבלה מהאינדקס הקטן ביותר (1) לגדול ביותר (n).

חילוץ פתרון: ניגש ונחזיר את הערך בטבלה $T[n]$.

4. <u>ניתוח זמן ריצה</u>:

בכל איטרציה האלגוריתם מבצע $O(1)$ עבודה, ואנו מבצעים n איטרציות, לכן האלגוריתם מבצע $O(n)$ עבודה.

שאלה 4:

נציע את האלגוריתם הבא:

1. נבצע מיון טופולוגי על הגרף, כך מהקודקוד משמאל מוציא צלע לקודקוד מימינו.
2. נגדיר טבלה בגודל $1 \times n$, ונשים את הערך $-\infty$ בכל התאים, למעט הקודקוד t, שם נשים את הערך 0.
   (a) במידה ויש קודקודים מימין ל-t במיון הטופולוגי, הם בהכרח לא יגיעו לקודקוד t, מהגדרת המיון הטופולוגי, ערכם ישאר $-\infty$
3. נגדיר את j להיות האינדקס המתאים לקודקוד t לאחר המיון הטופולוגי
4. נעבור על כל הקודקודים מ$j-1$ ל-1, ונשים את הערך בטבלה:

$$T[i] = max\left\{w(e_q) + \underset{(d(q))}{T[q]}; 1 \leqslant q \leqslant j-1\right\}$$

$$\forall e_q = (i,q) \in \underbrace{E_i}_{\text{קבוצת הצלעות היוצאות מהקודקוד ה}-i}$$

5. כלומר: המקסימום מבין האופציות להגיע לקודקוד t מקודקוד i.
6. נסיים את ריצת האלגוריתם, בכך בטבלה קיימים המרחק המקסימלי בין כל קודקוד ול-t.

פירוט עבור האלגוריתם הדינמי:

1. ניסוח בהיר של תתי הבעיות:
   נסתכל על המרחקים המקסימלים בין קודקוד a_i לכל $1 \leqslant i \leqslant n$ לקודקוד t.
2. נוסחת רקורסיה והסבר לבניתה:
   נוסחת הרקורסיה:

$$T[i] = \begin{cases} -\infty & i > t \\ 0 & i = t \\ \underset{\substack{q>i \\ (i,q)\in E_i}}{max}\left\{w(i,q) + \underset{=T[q]}{d(q)}\right\} & i < n \end{cases}$$

   הסבר: אם אנחנו בקודקוד t, נגדיר את המרחק המקסימלי להיות 0, לא ביצענו דבר
   אם קיימים קודקודי בור נוספים שלפי המיון הטופולוגי האינדקס שלהם גדול מהקודקוד t, הם יהיו $-\infty$ כמפורט באלגוריתם
   לכל קודקוד בעל אינדקס קטן יותר מהאינדקס של t, נחזיר את הערך להיות המקסימום מבין כל האופציות האחרות.
3. תיאור של הטבלה, סדר המילוי שלה ואופן חילוץ הפתרון:
   תיאור הטבלה: הטבלה בגודל $1 \times n$, כל תא מתאים לסידור החדש של המיון הטופולוגי, נמלא את הטבלה מהאינקס הגדול ביותר לקטן ביותר כפי שמפורט לעיל באלגוריתם
   אופן חילוץ המידע: הטבלה עצמה מכילה את המרחקים המקסימלים עבור כל קודקוד וקודקוד לכן נוכל להחזיר את הטבלה כרשימה.
4. ניתוח זמן ריצה:
   עבור כל קודקוד היות ואנחנו מתחילים מהקודקודים הקרובים ביותר לקודקוד t לאחר המיון הטופולוגי ומתרחקים ממנו לפי סדר המיון המפורט לעיל. עבור כל איטרציה: זמן הריצה הוא $O(n)$- מעבר על השכנים של הקודקוד הנוכחי, לכן עבור n קודקודים נבצע זמן ריצה של $O(n^2)$ עבור כל האלגוריתם