תרגיל בית מס' 4- גלי גלר 207704842

שאלה 1

סעיף א

מס הדרכים בהן ניתן לסדר את איברי הרשימה [1,2,...n] בשורה שקול לפתרון השאלה הקומבינטורית "בכמה דרכים שונות ניתן לסדר n אנשים בשורה?" ראינו בבדידה שהתשובה היא !n כי יש n אפשרויות לבחור את האדים הראשון ואז n-1 אפשרויות לבחור את השני עד שמגיעים למצב בו יש רק אדם אחד שעוד לא נבחר לו מקום, לכן גם מספר הדרכים בהן ניתן לסדר את איברי הרשימה הוא **!n**

סעיף ב

נרצה למצוא את מספר הסידורים של הרשימה עבורם הפונקציה det_quicksort תרוץ בזמן הארוך ביותר.

ראשית, נשים לב שהמקרה בו det_quicksort תרוץ בזמן הארוך ביותר הוא המקרה שבו smaller ריקה או greater ריקה, כלומר lst[0] הוא האיבר הגדול ביותר ברשימה או הקטן ביותר. נפתור בעזרת נוסחה רקורסיבית:

נתון כי n טבעי וחיובי, לכן מתקיים n > 0. עבור n = 1 יש רק אופציה אחת כי ברשימה יש איבר אחד, ועבור n = 2 יש שני איברים שונים לכן יש שתי אפשרויות שונות וסה"כ נקבל שתנאי העצירה הם:

$$a_1 = 1, a_2 = 2$$

נמצא את האיבר הכללי a_n :

לאיבר הראשון בסידור הרשימה יש שתי אפשרויות: הקטן ביותר או הגדול ביותר. נניח בלי הגבלת הכלליות שהאיבר שנבחר הוא הגדול ביותר. בצעד הבא נוכל לבחור את הקטן ביותר, שאחריו בעיה שקולה לסידור רשימה בגודל n-2. אחרת, נבחר באיבר השני בגודלו ברשימה, וגם במקרה זה הבעיה תהיה שקולה לסידור רשימה בגודל n-2.

לסיכום, נקבל: $a_n = 2(a_{n-2} + a_{n-2}) = 4a_{n-2}$

נפתור את נוסחת הנסיגה שקיבלנו בעזרת פולינום אופייני:

ראינו בבדידה שהפולינום האופייני יהיה $f(x) = x^2 - 4 \rightarrow$ $r_1 = 2, r_2 = -2$ נציב בנוסחאת הנסיגה עם תנאי העצירה:

$$1 = A * 2^1 + B * -2^1 = 2A - 2B$$
$$2 = A * 2^2 + B * -2^2 = 4A + 4B$$

לכן A = 0.5, B = 0 ולכן האיבר הכללי הוא: $a_n = \frac{1}{2}2^n = 2^{n-1}$

סה"כ קיבלנו כי מספר הסידורים של הרשימה עבורם הפונקציה תרוץ בזמן האיטי ביותר היא $\mathbf{2^{n-1}}$

סעיף ג

כעת נחשב את הגבול:

מהסעיפים הקודמים p(n) = n! ו w(n) = 2^{n-1}

$\lim_{n\to\infty}\frac{w(n)}{p(n)} = \lim_{n\to\infty}\frac{2^{n-1}}{n!}$ ומתקיים: $0 \le \frac{2^{n-1}}{n!} \le \frac{2^n}{n!} \le \frac{2}{1}*\frac{2}{2}*\frac{2}{n}$

כי לכל i בין 2 לn $0 < \frac{2}{i} < 1$ ולכן יקטין את המכפלה שבצד הימני ביותר באי השוויון

כמו כן, מתקיים:

$$\lim_{n\to\infty}\frac{2}{1}*\frac{2}{2}*\frac{2}{n} = 4\lim_{n\to\infty}\frac{1}{n} = 0$$

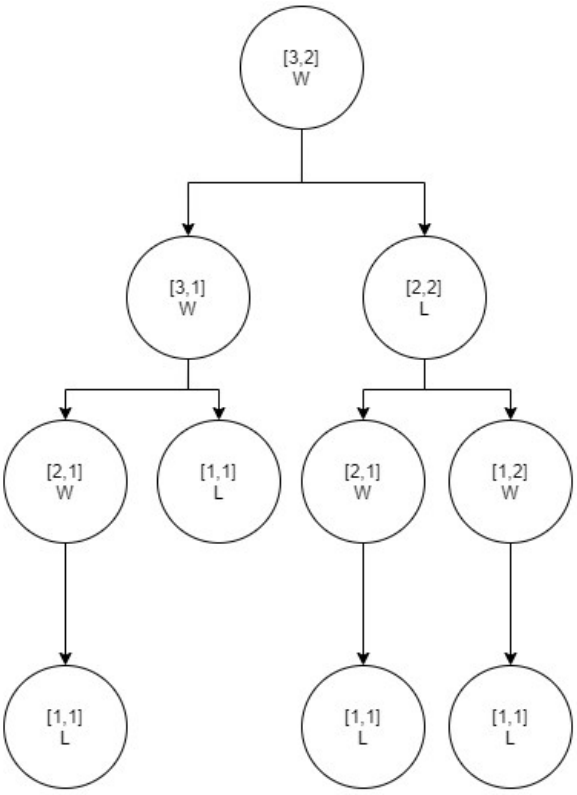
ולכן מכלל הסנדוויץ' שנלמד בקורס חדו"א1 מתקיים $\lim_{n\to\infty}\frac{w(n)}{p(n)} = \lim_{n\to\infty}\frac{2^{n-1}}{n!} = 0$

ניתן להסיק מכך שככל שn גדל הסיכוי לכך שזמן הריצה של האלגוריתם יהיה גרוע ביותר קטן

שאלה 2

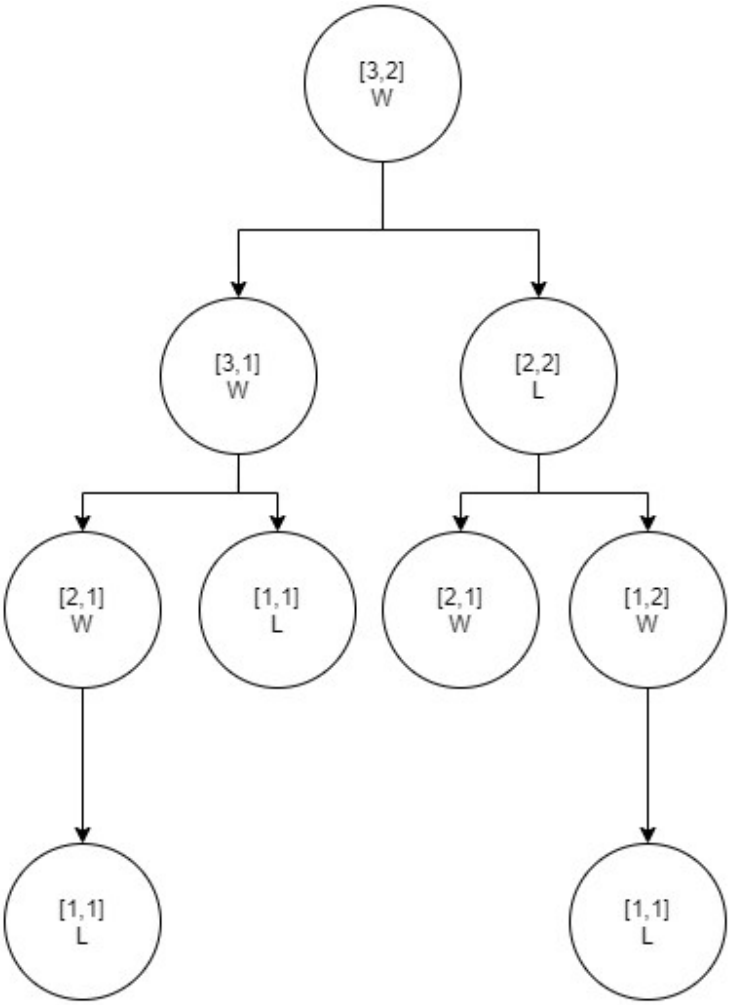
סעיף ב

הקונפיגורציה 1,1 מופיעה 4 פעמים בעץ



<u>סעיף ד</u>

הקונפיגורציה 1,1 מופיעה 3 פעמים בעץ



שאלה 3

סעיף א

נוכיח שלכל n במטריצה had(n) כל שורה מלבד השורה העליונה מכילה מספר זהה של אפסים ואחדות.

עבור n=0 זה מתקיים באופן ריק כי had(0) = [0] כי במטריצה יש רק שורה אחת.

נראה שזה אכן מתקיים עבור n=1:

$$had(1) = \begin{pmatrix} had(0) & had(0) \\ had(0) & \overline{had(0)} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

ואכן בשורה השנייה יש 0 אחד ו1 אחד כנדרש.

צעד האינדוקציה: נניח שעבור מטריצה בגודל *n* אכן מתקיימת התכונה, ונוכיח עבור מטריצה בגודל *n+1*:

מההגדרה של *had(n+1)* נקבל:

$$h(n+1) = \begin{pmatrix} had(n) & had(n) \\ had(n) & \overline{had(n)} \end{pmatrix}$$

קיבלנו מטריצת בלוקים שמורכבת מ*had(n)* ומ$\overline{had(n)}$. מהנחת האינדוקציה בכל שורה ב*had(n)* מלבד השורה הראשונה מספר ה1 שווה למספר ה0, ו2^n השורות הראשונות של *had(n+1)* הן "הכפלה" של השורות של *had(n)*, ולכן למעט השורה הראשונה מספר ה0 וה1 בהן שווה.

מההגדרה ב $\overline{had(n)}$ יש 0 איפה שב*had(n)* יש 1 ולהפך, לכן בשורה הראשונה של 2^n השורות האחרונות של *had(n+1)* יהיה את אותו המספר של 0 ו1, ומאותה סיבה היחס יישמר גם בשאר השורות.

לכן נסיק כי התכונה אכן מתקיימת לכל *n*

סעיף ג:

ננתח את סיבוכיות הקוד בהנחה ש *pow(2,n) = O(n)*:

```
def is_in_inverse(n, i, j):
    size_of_block = pow(2, n-1) # O(n)

    return (i >= size_of_block) and (j >= size_of_block) # O(1)

def had_local(n, i, j):
    if n == 0:
        return 0
    size = pow(2, n-1) # O(n)

    if is_in_inverse(n, i, j): # O(n)
        return 1 - had_local(n-1, i - size, j - size)
    if i >= size:
        i = i – size # O(1)

    if j >= size:
        j = j – size # O(1)

    return had_local(n-1, i, j)
```

כפי שניתן לראות, מלבד הקריאה ל *is_in_inverse* שהיא *O(n)* כל שאר הפעולות הן *O(1)*, ויתבצעו *n* קריאות לפונקציה לכן הסיבוכיות היא $\boldsymbol{O(n^2)}$

שאלה 5

מספר הביטים של *a* - *n*
מספר הביטים של *b* – *m*

כדי לחשב את הסיבוכיות נרצה לחשב כמה פעמים תרוץ לולאת ה*while* ולהכפיל במספר הפעולות שבתוך הלולאה.

ראשית, לולאת ה*while* תרוץ ***O(m)*** פעמים- בכל איטרציה של הלולאה מחלקים את *b* ב-2, ולכן הייצוג הבינארי של *b* מאבד ביט. ל*b* יש *m* ביטים ולכן הלולאה תרוץ *O(m)* פעמים

בתוך הלולאה מתרחשות 3 פעולות שסיבוכיות זמן הריצה שלהן גדולה מ*O(1)*: *result * a, a*a, b//2* והסיבוכיות של תוכן הלולאה היא חיבור של שלושת פעולות אלו.

נתון כי *b//2 = O(|b|)* ובכל איטרציה *|b|* קטן בביט אחד, לכן עבור

בנוגע לפעולות שקורות בתוך הלולאה, מהנתון ניתן להניח שכולן חוץ מפעולות הכפל והחילוק הן *O(1)*, כמו כן נתון כי *b//2 = O(m)*, לכן הסיבוכיות של כל הפונקציה היא *m*O(result * a + a*a)+* $\sum_{i=1}^{m} i$ כי מספר הפעולות הכולל שיתבצע עבור *b//2* הוא $\sum_{i=1}^{m} i = O(m^2)$

נחשב את הסיבוכיות של *a*a*:

כפי שניתן לראות בדוגמת הכפל למעלה, יתרחשו הכפלות של *a* בסדר גודל של $|a|^2$ - הכפלה של *a* בכל ביט של עצמו, והן יהיו מוזחות, לכן נקבל של *a*a* יש בערך *2|a|* ביטים, ויתרחשו פעולות חיבור בסדר גודל של *O(|a|²)*, לכן *a*a = O(|a|²)*

נשים לב כי $|result| \leq |a|$ בכל איטרציה, לכן $result * a = O(a*a)$

כעת נחשב את הסיבוכיות של $a*a$ כתלות בn, נשים לב כי באיטרציה הראשונה יתבצעו $O(n^2)$ פעולות ובאיטרציה השנייה יתבצעו $O((2n)^2)$ פעולות ובאופן כללי, אם i הוא מספר האיטרציה, יתרחשו $O(4^i n^2)$

נשים לב שכאשר $b=1$ הלולאה לא תתבצע, לכן מתרחשות $m-1$ איטרציות. נחשב את סכום סדרת הi:

$$\sum_{i=1}^{m-1} 4^i n^2 = n^2 \sum_{i=1}^{m-1} 4^i = n^2 \left(\frac{4-4^m}{1-3}\right) = O(n^2\, 4^m)$$

כעת נחשב את הסיבוכיות הכוללת:

$O(m(a*a + result * a + b//2)) = O(m(2*(n^2\, 4^m) + m^2)) = O(m(2 * (n^2\, 4^m) + 4^m)) = O(m(n^2\, 4^m)) = O(n^2\, 4^m)$

ולכן הסיבוכיות הכוללת היא $\boldsymbol{O(n^2\, 4^m)}$

<u>שאלה 6</u>

<u>סעיף ב</u>

סיבוכיות זמן הקוד שכתבתי היא **אקספוננציאלית**. הפונקציה קוראת לפונקציה שמחשבת את מרחק העריכה באמצעות אינדקסים, לכן כל הפעולות בפונקציה מלבד min הן $o(1)$, כמו כן min היא $o(n)$ ומקבלת תמיד קלט באורך 3, לכן נחשיב גם אותה כ$o(1)$, לכן הסיבוכיות תלויה רק בעץ הרקורסיה.

נשים לב שהמקרה הגרוע הוא המקרה שבו שתי המחרוזות שונות בכל מקום, כלומר התנאי

```
if s1[index_s1 - 1] == s2[index_s2 - 1]:
```

לא יתקיים עבור אף תו, כמו כן עבור n הרמות הראשונות אף פעם לא נגיע לתנאי העצירה כי בכל פעם האינדקס זז אחורה ב1, כלומר אנחנו מתייחסים למחרוזת כמחרוזת הקטנה בתו אחד מהקריאה הקודמת

לכן בn הרמות הראשונות של העץ לכל צומת יהיו שלושה בנים והסיבוכיות תהיה לפחות אספוננציאלית. כמו כן, עומק העץ המקסימלי הוא לכל היותר $2n$ – כי מספר הצמתים המקסימליים במסלול הוא כשאורך מחרוזת אחת נשאר קבוע והמחרוזת השנייה מתקצרת עד שמתקבלת מחרוזת באורך 1, ואז מתבצעת קריאה רקורסיבית שמקצרת גם את השניה למחרוזת ריקה, לכן הסיבוכיות המקסימלית היא לכל היותר אקספוננציאלית.

<u>סעיף ד</u>

לאחר הממואיזציה סיבוכיות זמן הריצה תהיה $\boldsymbol{O(n^2)}$

גם כאן כל הפעולות בגוף הפונקציה הן $O(1)$- גישה לתא ברשימה, פעולות אריתמטיות וmin שכאמור מקבל קלט בגודל קבוע- 3, לכן סיבוכיות זמן הריצה היא פונקציה של מספר הצמתים בעץ הרקורסיה כתלות בn.

נסתכל על הענף הראשון בעץ- הענף שלוקח את $s1$ ומשווה אותו לתת- מחרוזות הולכות וקטנות מ$s2$. תת העץ הזה ימלא תת-מטריצה מסדר $n*(n+1)$ ויבצע עוד מספר קריאות מסדר גודל $O(n)$ שכבר קיימות במערך או מגיעות לתנאי העצירה, סדר גודל סיבוכיות זמן הריצה הכולל של הענף הוא החיבור של שניהם, ולכן הוא יהיה $\boldsymbol{O(n^2)}$

כמו כן, כל שאר הענפים יבצעו פחות פעולות בגלל שהענף הראשון שומר ערכים במערך, ולכן יבצעו $O(n^2)$ או פחות קריאות לפונקציה.

לבסוף, הצומת הראשונה קוראת לכל שלושת ענפי העץ זה אחר זה, לכן יש חיבור בין סיבוכיות זמן הריצה של כל אחד משלושת הענפים, ולכן הסיבוכיות הכוללת היא $\boldsymbol{O(n^2)}$