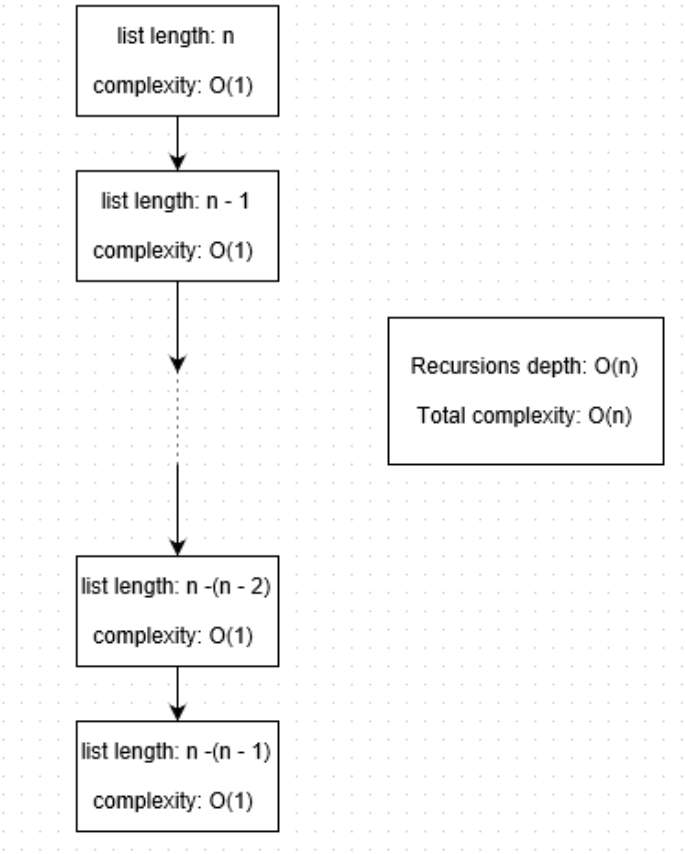
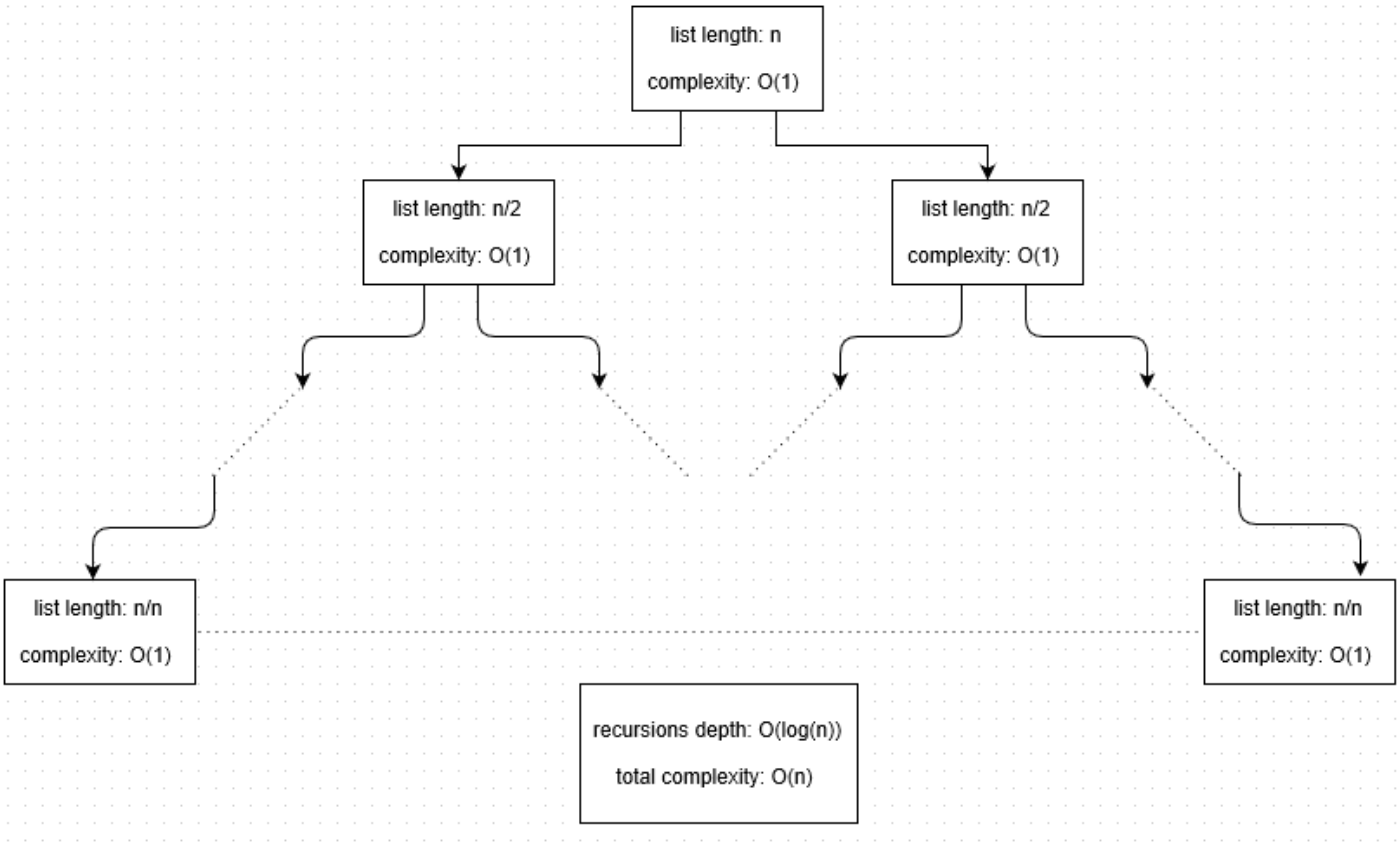
שאלה 1

א. עבור max11:



ג. עבור max22:



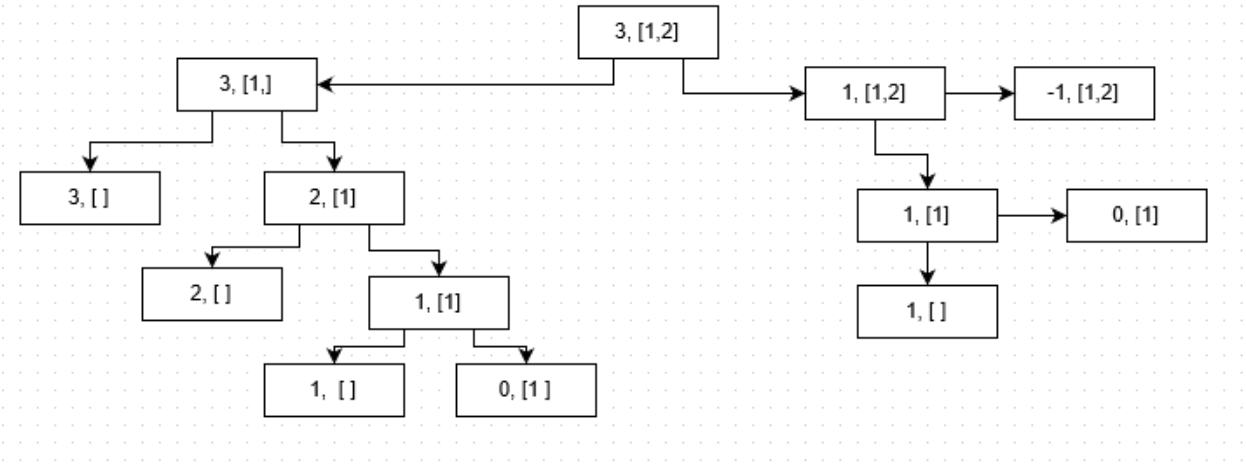
ד.

function	n = 1000	n = 2000	n = 4000
Max1	0.012137630760776119	0.0546660652501032	0.24298435866762702
Max 2	0.002181845682656558	0.004136269470620846	0.008888900893225582
Max_list11	0.0023001482833961973	0.0020149255747128336	0.004107639160906729
Max_list22	0.004726702085420698	0.005576428071719874	0.008409208155853776

ה. התוצאות המדדות מתיישבות יופי עם הסיבוכיות שחישבנו:
עבור max1, הסיבוכיות היא O(n**2), ואכן, כשהקלט גדל פי 2, הזמן גדל בערך פי 4.
עבור max2, הסיבוכיות היא O(n*logn), שזה בערך גידול לינארי, כפי שנראה בנתונים.
וגם עבור max11 ו- max22, שהסיבוכיות של שניהם היא O(n), אנו רואים שהגידול הוא ליניארי פחות או יותר.
<u>הערה:</u> אומנם רואים זאת רק במעבר מ2000 ל-4000 עבור השתיים האחרונות, אך הדבר נובע כנראה מטעויות מדידה של פייטון – במהלך המדידות קיבלתי גם קלטים אחרים שכן התאימו לתיאוריה. הכנסתי לטבלה את הקלט שהופיע הכי הרבה פעמים בממוצע.

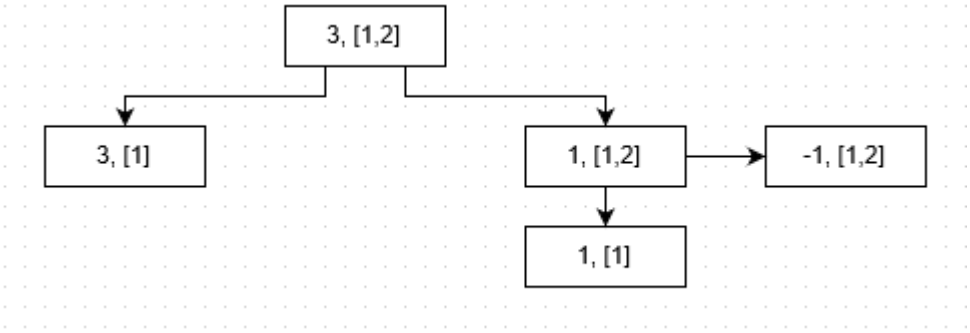
שאלה 2

א. גרף עבור הפונקציה change:

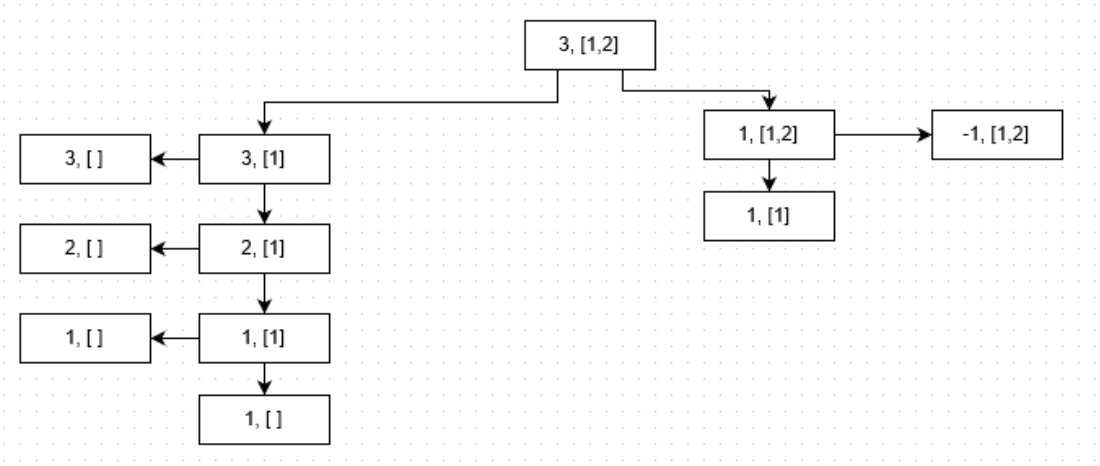


ב. גרף עבור chage_fast:

א) לפי הקוד שכתבתי:

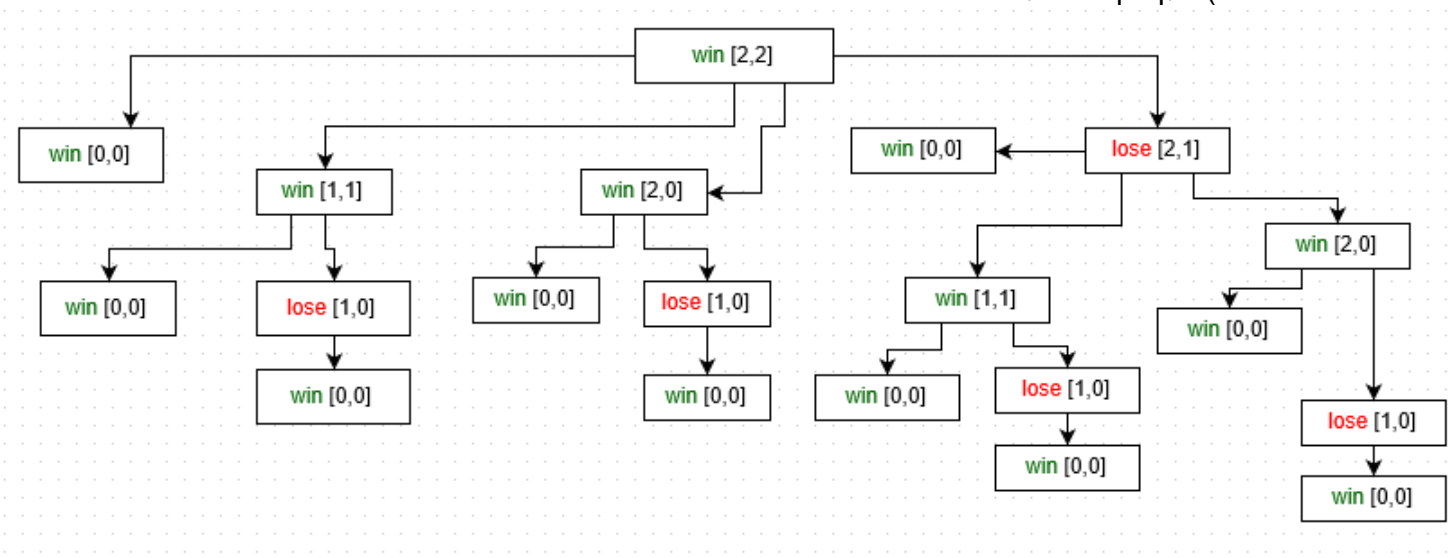


ב) לפי קוד עם Memoization בלבד (לא היה ברור לי מהסעיף למה כוונתכם).



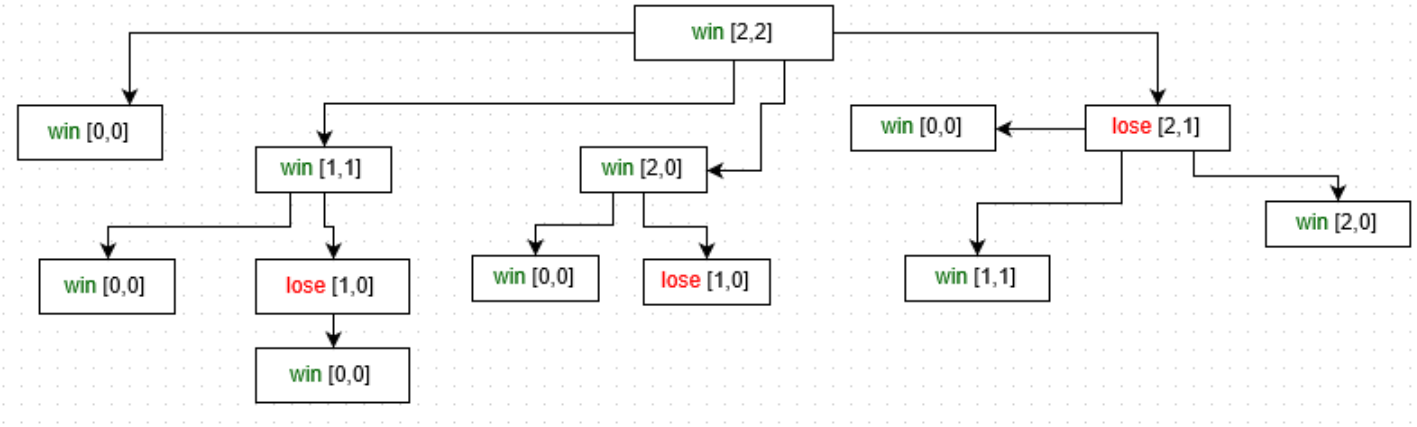
שאלה 3

א) עץ רקורסיה עבור win:



ב) עבור הביטוי שניתן, ה=n3 הוא הערך הגדול ביותר שרץ תוך פחות מדקה.

ד) עץ רקורסיה עבור win_fast:



ה) עבור הביטוי, n = 9 הוא הערך הגדול ביותר שנקבל שרץ תוך פחות מדקה.

שאלה 5

א)

Prime size (bits)	100	200	300	400	500
density	0.0139	0.0075	0.0055	0.0037	0.0027
Prime sentence	0.0142	0.0071	0.0047	----------	----------

לפי משפט המספרים הראשוניים, עבור X כלשהו, מספר המספרים הראשוניים שקטנים ממנו יהיה בקירוב $\pi(x) \approx \frac{x}{\ln x}$ (מוויקפדיה). כפי שניתן לראות מהטבלה, המדידות אכן מתאימות לתחזית של משפט המספרים הראשוניים.

ב) התשובה היא שתכניתו הנאלחת של דני **לא** תצליח, וגרוע מכך, שהוא אינו ראוי ללמוד מדעי המחשב.
הפונקציה מחזירה עד לפריקותו של מספר – והיא מתבססת על המשפט הקטן של פרמה, ועל כך שהוכח שעבור P פריק כלשהו, שלושת רבעי מהמספרים בטווח 1<a<p-1 יהיו עדים לפריקותו – כלומר, לא יקיימו את המשפט הקטן של פרמה.
בפשטות – עד של m הוא לא בהכרח מחלק שלו.
על אחת כמה וכמה כשמדובר במספר N עצום כלשהו שמתפרק רק לשני גורמים ראשוניים. הסיכוי שהעד שיחזור הוא בדיוק אחד מהמספרים האלה, הוא פשוט אפסי.
אז למרות שייתכן שאיכשהו הוא יקבל את שני הגורמים, יותר סביר שהוא יישאר עם שבר ביד.

שאלה 6

עם המידע שיש ברשותו יוכל הסטודנט לגלות את המפתח הסודי המשותף.

נניח שמתקיימים התנאים: x = (g**a)%p , y = (g**b)%p , key = (y**a)%p = (x**b)%p.

יהי a' המקיים x = (g**a')%p. נוכל להציב, ונקבל:
(ya')%p =** ((g**b)**a')%p = ((g**a')**b)%p = (((g**a')%p)**b)%p = **(x**b)%p = key**

כל המעברים מתאימים לחוקי mod.
כך מתקבל למעשה שתיאורטית, עם אתה מקבל a' כזה, תוכל לפצח את הקוד.

עם זאת, ראוי לציין שהשיטה עדיין נחשבת לבטוחה, שכן כאשר אתה מיישם אותה עם מספרים ראשוניים מאוד גדולים, הסיכוי שלך לחלץ a' כזה הוא אפסי עם הידע של היום.