תרגיל בית מס 3- גלי גלר 207704842

תרגיל מס 1

סעיף א

1. **הוכחה:** לכל n>1 מתקיים כי log n לכן מתקיים:

לכן אכן nlogn = O(n!)

1. **הוכחה:**

כי עבור n גדול מספיק מתקיים לכל i בין 0 ל k-1

לכן עבור c = מתקיים כי ולכן אכן

1. **הוכחה:** מהנתון כי ו קיים m טבעי ו כך שלכל m<n מתקיים ו , נסמן: c =

לכן אכן

1. **הפרכה:** נגדיר: f(n) = g(n) = h(n) = , בבירור f(n) וגם לכן אכן מתקיים f(n)=O(g(n)) וגם h(n) = O(g(n)) אך לכן אכן מתקיים:
2. **הוכחה:** מהנתון קיימים כך ש f(n) וg(n) לכן מתקיים:

f(n) , נסמן c = ונקבל כי f(n) לכן אכן f(n) = O(h(n))

1. **הוכחה:**

לכל n > 1 מתקיים: logn > 0 ועבור k מתקיים , כמו כן מתקיים לכן נשתמש בהגדרת הגבול, עם הגבול ( מכלל הסנדוויץ' חדוא1:n )

1. **a: *הפרכה:*** נגדיר: לכל i בין 1 ל n אז מתקיים כי max{}(n) = n ולכן O(max{}(n)) = O(n) אבל

**b: הפרכה:** נגדיר: לכל i בין 1 ל n-1 ו אז מתקיים: max{}(n) = n ולכן O(max{}(n)) = O(n) אבל

*סעיף ב*

1. *ראשית, ננתח את הסיבוכיות של כל פעולה בקוד:*

def f1(L):

n = len(L) **O(1)**

for i in range(n): **O(n)** – the loop runs n times

if i in L: **O(n)** – explanation below

L.append(i) **O(1)** - given

return L

הפעולה i in L עוברת על איברי L ומחפשת את i- במקרה שבו i לא נמצא יתבצעו len(L) מעברים  
כמו כן, len(L) משתנה כאשר i נמצא- לכן הפעולה append תקרה לכל היותר n פעמים, לכן len(L) הוא לכל היותר 2n וסהכ פעולה היא O(n)

הלולאה רצה n פעמים ועבור כל ריצה שלה מתבצעת פעולה שהסיבוכיות שלה היא O(n) הסיבוכיות של הפונקציה היא **O(**

1. *ננתח את הסיבוכיות של כל פעולה בקוד:*

def f2(L):

n = len(L) **O(1)**

res = [] **O(1)**

for i in range(500, n): **O(n)** – the loop runs n – 500 <= n times

for j in range(i,3\*i): **O(n) -** explanation below

k=1 **O(1)**

while k < n: **O(logn)** – explanation below

k+=2

res.append(k)

return res

הלולאה השנייה רצה 2i פעמים לכל i, כמו כן i < n לכן 2i <= 2n והסיבוכיות של הלולאה היא O(n)

ראינו בתרגול שהלולאה השלישית רצה בlogn

לכן סה"כ הסיבוכיות של הפונקציה היא **O(**

סעיף ג

הפונקציה הראשונה יוצאת רשימה חדשה המורכבת מעותק של הרשימה המקורית והאיבר החדש, בעוד שהפונקציה השניה משנה את הרשימה שהיא מקבלת כקלט.

כאשר אלחנן קורא לפונקציה הראשונה, לולאת הfor עוברת על הרשימה המקורית גם לאחר שהמשתנה l כבר לא מצביע עליה- הוא מצביע על הרשימה החדשה שהחזירה הפונקציה, ולכן הלולאה תרוץ רק עד סוף הרשימה המקורית.

לעומת זאת, כאשר אלחנן יקרא לפונקציה השנייה, היא תוסיף איברים לרשימה המקורית בכל איטרציה של הלולאה, ולכן הלולאה לעולם לא תגיע לסוף הרשימה.