תרגיל 3:

שאלה 1:

1) א. **הטענה לא נכונה:** $16^{\log n} = 2^{4\log n} = 2^{\log n^4} = n^4$ ולכן הביטוי אסימפטוטית גדול יותר ממשפחת הפונקציות $O(n^3)$.

ב.**הטענה לא נכונה:** $\lim \frac{n^2 \log n}{n\,(\log n)^2} = \lim \frac{n}{\log n} = \infty$ ולכן הביטוי אסימפטוטית גדול יותר מאשר משפחת הפונקציות $O(n\,(\log n)^2)$.

ג. **הטענה נכונה:** למדנו בשיעור כי כאשר אנו מחברים ביטויים יחדיו אז הביטוי החזק ביותר הוא הביטוי המייצג מבחינה אסימפטוטית, לכן $(\sum f_i)(x) = O(\max\{f_i\})$ ומנתון אנחנו יודעים כי לכל f קיימת g שהיא O שלה, לכן בפרט בעבור ה-f המקסימלית קיימת g כזו, וה-g הזו נמצאת בסכום פונק' ה-g, ומכיוון שאנו מדברים על פונק' מהטבעיים לטבעיים בלבד אין לנו פונקציות שעלולות לבטל זו את זו, לכן $(\sum f_i)(x) = O(\max\{f_i\}) = O((\sum g_i)(x))$

ד. **הטענה נכונה:** בזהה לטיעונים של סעיף ג. אנו מקבלים $(\sum g_i)(x) = O(\max\{g_i\})$ ובשילוב המסקנה מסעיף ג. נקבל כי $(\sum f_i)(x) = O(\max\{g_i\})$

ה. **הטענה נכונה:**

$$f_1(x)\cdot f_2(x) \le c_1\cdot g_1(x)\cdot c_2\cdot g_2(x) = c_1\cdot c_2\cdot\left(g_1(x)\cdot g_2(x)\right) = O(g_1(x)\cdot g_2(x))$$

ו. **הטענה לא נכונה:** נראה דוגמא נגדית, $2^n = O(2^n), 2n = O(n)$ אבל אם נרכיב את הפונק' נקבל $2^{2n} = 4^n$ וקל לראות כי הרכבה של פונק' ה-*O* בסדר זהה משמר את 2^n מה שכלל איננו בקטגוריה של $O(4^n)$.

ז. **הטענה לא נכונה:** נראה כי הביטויים נבדלים בסדר גודל לוגריתמי לכל שני קבועים שנבחר:

$$n^n \le c\cdot b^{nt} \leftrightarrow e^{n\ln n} \le e^{\ln c + nt\ln b}$$

אבל נשים לב כי:

$$\lim\frac{\ln c + nt\ln b}{n\ln n} = \lim\frac{\ln c}{n\ln n} + \lim\frac{nt\ln b}{n\ln n} = 0 + \lim\frac{t\ln b}{\ln n} = 0$$

לכן לכל שני קבועים שהם הביטוי n^n תמיד יהיה אסימפטוטית גדול יותר מאשר b^{nt}.

2) נוכיח כי $(2^k+\varepsilon)^{\log n} \ne O(n^k)$
נשים לב כי:

$$n^k = 2^{k\log n} = (2^k)^{\log n}$$

לכן:

$$\lim\frac{n^k}{(2^k+\varepsilon)^{\log n}} = \lim\frac{(2^k)^{\log n}}{(2^k+\varepsilon)^{\log n}} = \lim\left(\frac{2^k}{2^k+\varepsilon}\right)^{\log n} = 0$$

זאת מכיוון שכאשר אנו מעלים גורם בטווח *(-1,1)* בחזקה ששואפת לאינסוף אנחנו תמיד נשאף ל-0.
לכן הביטוי $(2^k+\varepsilon)^{\log n}$ אסימפטוטית גדול יותר מ-n^k.

3) *א. קוד:*

```
def f1(L):
    n = len(L)
    while n > 0:
        n = n // 2
        for i in range(n):
            if i in L:
                L.append(i)
    return L
```

תשובה: $O(n^2)$

הלולאה החיצונית רצה *log(n)* איטרציות והלולאה הפנימית רצה בכל פעם $\frac{1}{2^k}n$ איטרציות כאשר בכל איטרציה *in* רצה בסיבוכיות $O(n)$, סה"כ:

$$n^2 \sum_{k=1}^{\log n} \frac{1}{2^k} = n^2 \left(\frac{1 - 0.5^{(\log n)+1}}{0.5} - 1 \right) = n^2 \left(1 - 0.5^{\log n} \right) = O(n^2)$$

ההבדל בין המקרה הטוב למקרה הגרוע של הפונק' היא פקודת *append* בכל איטרציה פנימית, אבל מכיוון בלאו וכי אנו מבצעים פעולות אחרות בדרך לפקודה הזו (הלולאה עצמה, התנאי) זמן הריצה של הפונק' הוא $O(n^2)$ כך או כך.

ב. קוד:

```
def f2(L):
    n = len(L)
    res = []
    for i in range(500, n):
        m = math.floor(math.log2(i))
        for j in range(m):
            k=1
            while k<n:
                k*=2
                res.append(k)
    return res
```

תשובה: $O(n\,(\log n)^2)$

הלולאה החיצונית מבצעת בערך *n-500* איטרציות ולכן בסה"כ *O(n)* איטרציות.

הלולאה השנייה מבצעת $O(\log(i))$ איטרציות כאשר *i* מייצג את האינדקס הנוכחי.

הלולאה הפנימית ביותר מבצעת $O(\log(n))$ איטרציות.

סה"כ יש לנו:

$$\sum_{i=500}^{n} \sum_{j=1}^{[\log i]} \log n = \log n \cdot \sum_{i=500}^{n} \sum_{j=1}^{[\log i]} 1 = \log n \cdot \sum_{i=500}^{n} [\log i]$$

אז נסב תשומת ליבנו לביטוי הבעייתי ונתעלם מהעיגול כלפי מטה לנוחייתנו, במקרה זה אין הדבר משפיע על ההערכה האסימפטוטית (נראה כי הביטוי חזק יותר אסימפטוטית מ-*n*)

$$\left(\sum_{i=500}^{n} \log i \right) - (n - 500) = \sum_{i=500}^{n} \log i - 1 \leq \sum_{i=500}^{n} [\log i] \leq \sum_{i=500}^{n} \log i$$

כעת משהפרדנו החוצה ומכיוון שהפונקציה הפנימית מונוטונית עולה נחסום באמצעות אינטגרל:

$$\int_{499}^{n} \log x \, dx \leq \sum_{i=500}^{n} \log i \leq \int_{500}^{n+1} \log x \, dx$$

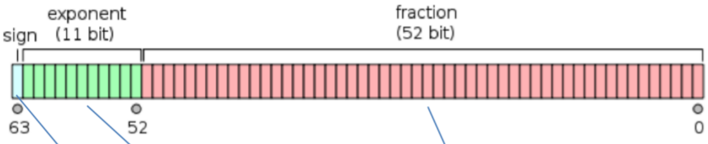
נחסוך פה את פתיחת האינטגל ורק נגיד כי לצרכינו, האינטגרל המסויים בקטע *[c,n]* כאשר *c* קבוע כלשהו ו-*n* שואף לאינסוף מתנהג כמו $O(n \log n)$.

כעת נכפול ב-$\log n$ החיצוני שנותר ונקבל כי זמן הריצה של הפונק' הוא מסדר $O(n\,(\log n)^2)$.

שאלה 2:

1) נראה זו בצורה פשוטה:

ייצוג 64 ביט:



(figure from Wikipedia)

and is coded by:

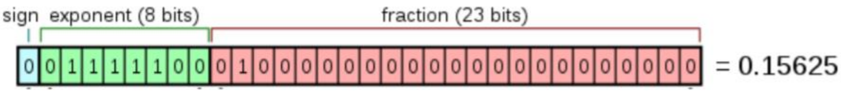
$$\boxed{(-1)^{sign} \cdot 2^{exponent-1023} \cdot (1 + fraction)}$$

ייצוג 32 ביט:

והנוסחה לחישוב המספר היא:

$$\boxed{num = (-1)^{sign} \cdot 2^{exp-127} \cdot (1 + fraction)}$$

דוגמה להמחשה:



תחילה נשים לב כי חישוב החלק השברי הוא זהה בעבור שני הייצוגים:

$$fraction = \sum_{i=1}^{n} b_{i+exp} 2^{-i}$$

כאשר *n* מייצג את הייצוג (64 \ 32) ו-*exp* מייצג את מספר הביטים שדורש הייצוג של האקספוננט.

לכן, כל שבר שניתן להציג בייצוג של 32 ביט ניתן להציג גם ב-64 ביט.

כעת נתייחס לחלק האקספוננט.

בייצוג של 32 ביט ניתן להציג את טווח החזקות 2^{-127} עד 2^{128}, וב-64 ביט ניתן להציג את טווח החזקות 2^{-1023} עד 2^{1024} , גם במקרה זה אנו מכילים את כל הערכים של הייצוג ב-32 ביט.

וכמובן שמנגנון הסימון זהה בשני הייצוגים.

סה"כ הייצוג ב-64 ביט מכיל כל ערך אפשרי בייצוג 32 ביט.

2) להלן מספר בייצוג 64 שלא ניתן להציג ב-32 ביט:

החלק השברי הוא 28 אפסים ו-24 אחדות:

<u>1111111111111111111111110000000000000000000000000000</u>

החלק האקספוננציאלי הוא 0 אחד ו-10 אחדות:

<u>01111111111</u>

הסימן חיובי ולכן הביט הוא 0:

<u>0</u>

סה"כ:

$$num = (-1)^0 \cdot 2^{1023-1023} \cdot \left(1 + \sum_{k=1}^{24} \frac{1}{2^{-k}}\right) = 1 + \sum_{k=1}^{24} \frac{1}{2^{-k}} \in [1,2)$$

נשים לב כי לא ניתן לייצג את החלק השברי ב-32 ביט פשוט בגלל שהחלק השברי של ייצוג זה מכיל 23 ביטים בלבד ובהתאמה מסוגל להגיע לדיוק של עד כדי $\frac{1}{2^{-23}}$.

3) קוד
4) קוד
5) קוד

6) א. ניתן לייצג 2^{23} מספרים שונים בטווח $[2^k, 2^{k+1})$.

נסביר על דרך השלילה,
ביט הסימן איננו בשימוש מכיוון שהסימן בטווח נשאר קבוע.
8 ביטים בעבור האקספוננט חייבים להישאר זהים אחרת "יזיזו" את k.
לכן נשארו רק 23 ביטים אשר בעבור כל סידור של 9 הביטים שפרטנו לעיל, פורשים את אותן 2^{23} קומבינציות בעבור הטור ההנדסי $\sum_{k=1}^{23}\frac{1}{2^k}$ בתוך טווח הולך וגדל בצורה אקספוננציאלית.

ב. נשים לב כי $[3,10]$ מכיל את $[2,4),[4,8),[8,16)$ באופן חלקי או מלא.
מכיוון ש-$[4,8)$ מוכל במלואו אנחנו יודעים כי יש בטווח לפחות 2^{23} מספרים.
כעת נחשב את הקטעים החלקיים, לשם כך נשים לב:

$$3 = 2 + 1 = 2\cdot\left(1+\frac{1}{2}\right)$$

$$10 = 8 + 2 = 8\cdot\left(1+\frac{1}{4}\right)$$

אז אנחנו יודעים להגיד כי בקטע $[3,4)$ יש 2^{22} מספרים אפשריים
ובקטע $[8,10]$ יש 2^{21} מספרים אפשריים, לכן בסה"כ יש בקטע כולו:
$2^{23}+2^{22}+2^{21}$ מספרים אפשריים.

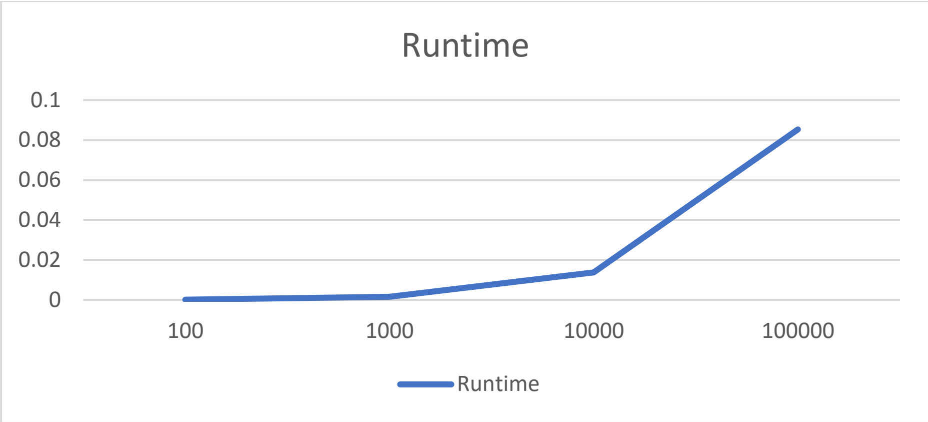
ג. בקטע $[0,1)$ יש $2^{30}-2^{23}$ מספרים אפשריים.
נסתכל על 7 הביטים הראשונים של האקספוננט כאשר הביט השמיני כבוי וביט הסימן כבוי (משמע חיובי), כאשר כולם דולקים אנו נמצאים במצב של $2^0 = 1$ ובכל מצב אחר אנחנו באקספוננטים שליליים – משמע שברים חיוביים.
למעשה בדומה לטיעוני סעיף קודם כל הקטעים $\left[\frac{1}{2^{-127}},\frac{1}{2^{-126}}\right)$, ... $\left[\frac{1}{4},\frac{1}{2}\right)$, $\left[\frac{1}{2},1\right)$ מוכלים בקטע $[0,1)$ בייצוג של 32 ביט.
אז אם נקזז את ייצוג ה-0 עם ייצוג ה-1 נשאר עם 127 קטעים שבכל אחד יש 2^{23} מספרים אפשריים, סה"כ: $2^{23}\cdot 127 = 2^{23}(2^7-1) = 2^{30}-2^{23}$

ד. מכיוון שלכל הטווח $[32,64)$ אנחנו נאלצים להתחיל עם 2^5 באקספוננט, אנחנו מצמצמים את הדיוק המקסימלי שלנו מ-2^{-23} ל-2^{-18}.

שאלה 3:

2)

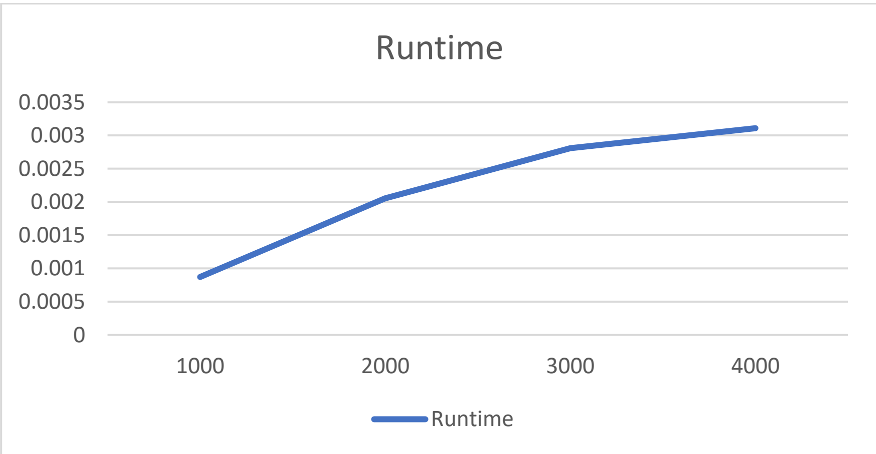
Iterations	*Runtime (seconds)*
100	0.000111
1000	0.001500
10000	0.013757
100000	0.085319



מוטב לא להתבלבל מהגרף, שכן אמנם זה נדמה כגרף פולינומיאלי אבל גם הקלטים שלנו גדלים בכל פעם בסדר גודל, לכן אנחנו ככל הנראה רצים בסדר גודל לינארי, (ואילו הגרף היה נראה לינארי היינו מסדר לוגריתמי).

לצורך הדגמה נראה את זמני הריצה על סדרת קלטים חשבונית:

Iterations	*Runtime (seconds)*
1000	0.000872
2000	0.002057
3000	0.002809
4000	0.003110



<u>המקרה הגרוע ביותר:</u>

יכול להיות שלא הבנתי את השאלה, כי מדובר פה על זמן ריצה הסתברותי, לדוגמא במקרה הלא סביר שבריצה מסויימת כל המספרים הרנדומיים יהיו קטנים מ-0.1 נקבל זמן ריצה שהוא בסדר גודל $10 \cdot N$ ואולי אפילו יותר גרוע בעוד שהממוצע שואף לזמן ריצה מסדר גודל $e \cdot N$.

אבל בכל מקרה, גם במקרה הטוב וגם במקרה הגרוע, אנחנו מסדר $O(N)$.

<u>הערה:</u> במידה ואנחנו מדברים על משתנים ללא הגבלת דיוק כפי שצויין בתחילת השאלה, אנחנו עלולים להגיע למצב שבו זמן הריצה אינסופי, למשל, סכום של חזקות של $\frac{1}{2}$ שואף ל-1 אבל אף פעם לא שווה ל-1.

שאלה 4:

1) סיבוכיות זמן הריצה היא $O(\log n)$, יש לנו מספיק אינפורציה על הסדר של הרשימה בשביל לחפש בה בצורה בינארית.
2) מיון של רשימה כמעט ממויינת הוא די ישיר, בודקים לכל איבר האם הוא גדול יותר מהאיבר הבא אחריו, ואם כן מחליפים. סדר גודל זמן הריצה הוא $O(n)$. הסיבה שזה עובד היא מכיוון שכל איבר ימצא לכל היותר באינדקס שלימינו או שלשמאלו, וכאשר אוכפים תנאי זה על כל הרשימה בצורה אינדוקטיבית מהאיבר הראשון מגלים שהתנאי הוא אף יותר מגביל – האיבר הראשון ימצא רק באינדקס הראשון או השני, אם הוא בראשון אז אנחנו בחזרה באותה בעיה בגודל *n-1* אחרת האיבר היחיד שיכול להמצא במקומו באינדקס הראשון הוא האיבר השני, נסדר אותם וקיבלנו את אותה הבעיה בגודל *n-2*. המקרה הטוב ביותר שיכול להיות לבעיה הזו הוא $\frac{1}{2}\cdot n = O(n)$
3) ניתן להראות כי ההגדרה תמיד מתקיימת לאינדקס אחד לפחות ברשימה בצורה אינדוקטיבית – פרט למקרה המנוון של רשימה באורך 1, נסתכל על האיבר הראשון, אם הוא מינימום מקומי, ניצחנו, אם לא אז הוא גדול יותר מהאיבר השני, שוב, אם האיבר השני מינימום מקומי, ניצחנו, אם לא אז הוא גדול יותר מהאיבר השלישי. מכיוון שהרשימה סופית ובמידה ולא מצאנו מינימום מקומי, קיבלנו רשימה מונוטונית יורדת והאיבר האחרון הוא בהכרח מינימום מקומי.
סיבוכיות: $O(n)$
אין לנו שום הנחה מקלה על מבנה הרשימה, וההגדרה היא לוקאלית לכל איבר והשכנים המיידיים שלו, עלינו לעבור על כל איבר, אחרת קיימים מצבים בהם אנחנו עלולים לפספס.

שאלה 5:

סיבוכיות פונק' ראשונה: $O(nk+5^k)$

בתחילת הפונק' אנחנו מאתחלים לולאה באורך 5^k באפסים.
אנחנו עוברים בלולאה על כל איבר ברשימה פעם אחת וקוראים לפונק' מסעיף א' הממירה מחרוזת למספר שלם. הפונק' מסעיף א' היא מסיבוכיות $O(k)$ כאשר k היא אורך המחרוזת.
לכן בסה"כ הלולאה שלנו היא מסיבוכיות $O(nk)$.
בסוף אנחנו עוברים על הרשימה בת 5^k איברים וקוראים לפונק' מסעיף ב' סה"כ n פעמים, פעם אחת בשביל כל איבר ברשימה. גם הפונק' הזו היא מסיבוכיות זהה לפונק' מסעיף א'.
לכן הלולאה האחרונה היא מסיבוכיות $O(nk+5^k)$.
סה"כ יש לנו חיבור סיבוכיות מהצורה:

$$O\left(5^k\right)+O(nk)+O\left(nk+5^k\right)=O(nk+5^k)$$

כרצוי.

סיבוכיות הפונק' השנייה: $O(nk\cdot 5^k)$

אנחנו מאתחלים רשימה ריקה, וזו הרשימה היחידה שאנחנו מחזיקים בהתאם להוראות התרגיל.

יש לנו שתי לולאות מקוננות –

1. לולאה חיצונית שעוברת על כל המספרים מ-0 עד 5^k-1, סה"כ יש לנו 5^k איטרציות.
2. לולאה פנימית שעוברת על כל המחרוזות ברשימה, ממירה אותן למספר טבעי בעזרת פונק' מסעיף א' ומשווה אותן לאינדקס הנוכחי של הלולאה החיצונית, כמו שהראנו בניתוח הסיבוכיות של הפונק' הראשונה מדובר על לולאה עם סיבוכיות של $O(nk)$.

מכיוון שהלולאות מקוננות, על מנת לחשב את הסיבוכיות של שתיהן יחדיו עלינו להכפיל את הסיבוכיות שלהן, סה"כ $O(nk\cdot 5^k)$