



## קורס מבני נתונים

### תרגיל תכנות מס' 1

#### קבוצה 15

##### מגשים:

תומר מוצרי 206782641

זוהר עציץ 318590791

פלג עז-ארי 314632019

**שאלה 1:** מימוש Merge sort

מימוש קוד כולל התייעוד נמצא בקובץ q1\_15

**שאלה 2:** מימוש Insertion sort

מימוש קוד כולל התייעוד נמצא בקובץ q2\_15

**שאלה 3:** השוואת חלופות

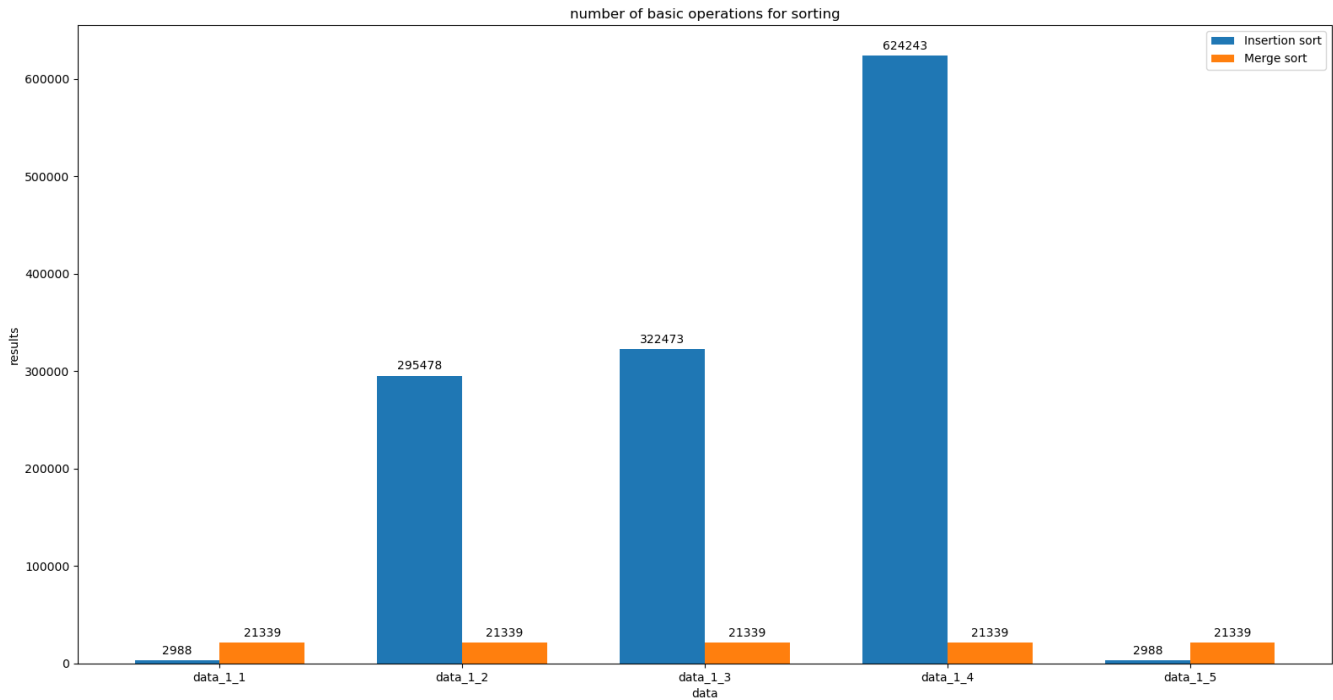
מימוש קוד כולל התייעוד נמצא בקובץ q3\_15

החבילות Python שהיו בשימוש לטובת מימוש הקוד הם:

- Pandas – לטובת ייבוא נתונים וקריאתם מקובץ האקסל.
- Numpy – לטובת יצירת הגרף.
- Matplotlib – לצורך תצוגה ועיצוב הגרף.



## הגרף שהתקבל מיצירת הקוד:



הסבר לגרף בהתאם לסוג המיון:

### Insertion sort

נחלק את התנהגות הקוד למקרי קיצון:

**המקרה הטוב ביותר** - במקרה זה הקלט שמתקבל הוא בצורה ממוינת בסדר עולה.

במקרה זה בהרצת הקוד תהיה כניסה ללולאת ה for אך לא ללולאת ה while, לכן על כל איבר יבוצעו מספר פעולות קבוע ללא צורך בסידור מחדש.

במקרה שלנו המקרה הטוב ביותר הוא בקבצים 1 ו- 5 - הממוינים בסדר הנכון.

על פי הגרף שיצרנו ניתן להבחין כי אכן בקבצים אלו מספר הפעולות הן הקטנות ביותר ביחס לקבצים האחרים, בנוסף מספר הפעולות קטן יותר מה-merge sort מכיוון שמיון זה "מזהה" שהרשימה מסודרת ומבצע רק פעולות של בדיקה.

**המקרה הגרוע ביותר** - במקרה זה הקלט שמתקבל הוא בצורה הממוינת בסדר יורד.

במקרה זה בהרצת הקוד תהיה כניסה ללולאת ה for וגם ללולאת ה while, לכן על כל איבר כמות הפעולות שצריך לעשות עולה ככל שמתדמים ברשימה, ולכן כמות הפעולות גדלה בצורה מעריכית. במקרה שלנו המקרה הגרוע ביותר הוא בקובץ 4 - הממוין בסדר יורד.



על פי הגרף שיצרנו ניתן להבחין כי אכן בקובץ זה מספר הפעולות הוא הגבוה ביותר ביחס לקבצים האחרים, בנוסף מספר הפעולות גדול בהרבה מה-merge sort.

**מקרה ביניים-** במקרה זה הקלט שמתקבל הוא ממיון בצורה רנדומלית. במקרה זה בהרצת הקוד תהיה כניסה מלאה ללולאת ה for וכניסה חלקית ללולאת ה while. במקרה שלנו מקרי הביניים הם בקבצים 2 ו-3- הממוינים בסדר רנדומלי. על פי הגרף שיצרנו ניתן להבחין כי אכן בקובץ זה מספר הפעולות הוא תלוי במידת הסידור של רשימת הקלט.

#### **merge sort:**

על פי מה שלמדנו הכיתה פונקציית merge sort היא עושה פעולה של "הפרד ומשול". פעולת המיון מתבצעת באמצעות חילוק למיון חציו הראשון של המערך ומיון חציו השני, ופתירת כל אחד מהם באמצעות קריאה רקורסיבית עד לבעיה של רשימה באורך 1 – הפרד. בתום פתרון תתי-הבעיות, האלגוריתם ממזג את שני חצאי המערך, הממוינים כעת, למערך ממיון גדול המכיל את כל איבריהם, משול. וזהו בדיוק פתרון הבעיה.

עבור כל ערך של הקלט תהליך המיון הוא אותו תהליך ולכן אין חשיבות לסדר הרשימה שמתקבלת, אלא רק לאורך הרשימה. (כמובן שיכולים להתקיים מקרים בהם מספר הפעולות יהיה שונה, אך ההבדלים הם זניחים) על פי הגרף שיצרנו אכן ניתן לראות שמספר הפעולות המתקבל הוא זהה בין כל הרשימות בשיטת מיון זו.

מספר הפעולות הבסיסיות שמתבצעות בתהליך המיון של המקרה הגרוע ביותר קטן משמעותית מה-insertion sort, מכיוון שככל שמספר האיברים ברשימה גדל, מספר הפעולות שנדרשות בתהליך מיון זה גדל בצורה לוגריתמית, בהשוואה לגדילה מעריכית של שיטת המיון הקודמת. ואכן על פי הגרף ניתן לראות הבדל משמעותי במקרה הגרוע ביותר וכן גם במקרים בהם הרשימה מסודרת בסדר רנדומלי.

#### **שאלה 4:** מימוש Selection sort

מימוש קוד כולל התייעוד נמצא בקובץ q4\_15