**מבני נתונים ואלגוריתמים:**

1. כתבו פונקציה המקבלת 2 רשימות ומחזירה רשימה חדשה המכילה רק את האיברים המופיעים בשתי הרשימות.(חיתוך)
2. כתבו פונקציה המקבלת 2 רשימות ומחזירה רשימה חדשה המכילה את כל האיברים שמופיעים רק ברשימה אחת(הפרש)
3. כתבו פונקציה המקבלת רשימת מספרים וממיינת אותה בעזרת Bubble sort
4. כתבו פונקציה המקבלת רשימת מספרים וממיינת אותה בעזרת Merge sort
5. כתבו פונקציה המקבלת רשימת מספרים ממוינת ומספר ומבצעת חיפוש בינארי כדי לבדוק האם קיים ברשימה.
6. כתבו פונקציה המקבלת רשימת מספרים ומחזירה את סכום תת הרשימה הרציפה הגדולה ביותר.
7. כתבו פונקציה המקבלת מחרוזת ומחזירה כמה מופעים יש לכל תו במחרוזת.
8. כתבו פונקציה המקבלת רשימה והופכת אותה למילון כך שהאינדקס של המילה ברשימה יהיה המפתח והערך של המילה יהיה הערך.

**OOP:**

1. בנו מחלקה BSTNode המייצגת קודקוד בעץ חיפוש בינארי המכילה את השדות הבאים:
   1. ערך הקודקוד – יהיה משתנה מטיפוס כללי
   2. בן ימני
   3. בן שמאלי
2. בנו את הקונסטרקטור באופן הבא:  
   בעת בניית קודקוד, חובה להזין את הערך של הקודקוד.  
   הערך הדיפולטיבי של הבן הימני והשמאלי הוא None, כלומר לא חובה על המשתמש להכניס כקלט את הבן הימני או הבן השמאלי.
3. הוסיפו למחלקה גטרים וסטרים לשדות
4. בנו מחלקה של עץ חיפוש בינארי BST התומכת בפונקציות הבאות:
   1. הוספת איבר לעץ
   2. חיפוש איבר בעץ
   3. מחיקת איבר מהעץ(להחלטתכם האם האיבר שיחליף קודקוד יהיה העוקב לו או הקודם לו ולתקן את העץ בהתאם לשינוי שבוצע)
5. הוסיפו למחלקה פונקציה השומרת לקובץ Json את האובייקט בצורה כזאת שנוכל לקרוא את האובייקט ולבנות בדיוק את אותו העץ  
   רמז: כנראה תצטרכו פונקציה נוספת שתעזור לכם לאחד החלקים.
6. צרו למחלקה איטרטור העובר על העץ לפי inOrder
7. הוסיפו factory method המקבלת מחרוזת המייצגת סוג מעבר ומחזירה איטרטור שעובר על הBST לפי אותו מעבר.  
   סוגי המעברים שעליכם לתמוך בהם:
   1. Inorder
   2. Preorder
   3. Postorder
   4. levelOrder
   5. reversedLevelOrder

שם המעבר הוא המחרוזת הדרושה לכם על מנת לבחור את האיטרטור המתאים, תוכלו לבחור אחד דיפולטיבי מבין החמישה.