פרויקטעציחיפושבינאריים

תקציר

בפרויקט זה נתחנו מספר גרסאות שונות של עצי חיפוש בינאריים, במקרה שקיימים בעץ צמתים בעלי מפתחות זהים. במטרה

למצוא את הגרסא הטובה ביותר. על מנת למצוא את הגרסה האופטימאלית מבחינת סיבוכיות זמן הריצה, השווינו בין העצים

השונים ע"י מנית מספר ההשוואות בעת הכנסת צומת חדש לעץ.

תוצאות המחקר היו חד משמעיות העץ הטוב ביותר הינו העץ שמשרשר צומת בעלת מפתח זהה למפתח קיים באמצעות רשימה

מקושרת (להלן "Tree b"). ואלו העץ הגרוע ביותר הוא העץ חיפוש הסטנדרטי (להלן "Tree").

ואילו כאשר נכניס איברים בעלי מפתחות שונים ללא חזרות מספר ההשוואות יהיה זהה בכל סוגי העצים.

בפרויקט הנ"ל ממשנו באופן מעשי את מבני הנתונים השונים בשפת python. התוכנית הורצה ב- PyCharm 2019.1.2

סעיףא':

1.

כאשר נכניס לעץ חיפוש בינארי n מפתחות זהים העץ יראה באופן הבא:

על פי ההגדרה שכאשר נכניס צומת עם מפתח זהה, למפתח



**root**

של צומת שכבר קיים בעץ פונים ימינה

סיבוכיות זמן הריצה הכוללת**:**

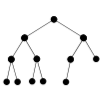
סיבוכיות הזמן נובעת מכך שעוברים עלn איברים, סיבוכיות הזמן של Tree\_insert בכל הכנסה של צומת חדש היא

כי העץ נראה כמו רשימה מקושרת.

אם נשפר את הגרסה הסטנדרטית ע"י שכאשר נכניס צומת לעץ ובמהלך הירידה בעץ ע"מ למצוא את מקום הצומת

החדש נתקל בצומת עם מפתח זהה נפנה ימינה או שמאלה לחילופין (להלן "Tree a") כל קומה בעץ תתמלא לגמרי

לפני שנרד לקומה הבאה. העץ יראה באופן הבא:



**root**

סיבוכיות זמן הריצה הכוללת:

ניתן לשפר את העץ ולחסוך בכמות ההשוואות בהמשך..

2.

גרסה א' (Tree): היא גרסה דיפולטיבית עבור עצי חיפוש בינאריים, כאשר נכנס צומת בעל מפתח זהה פונים ימינה

ולשם הוא נכנס.

במקרה הגרוע נכתב לעיל

גרסה ב' (Tree\_a): בכל צומת קיים שדה נוסף בשם dir שמכיל את הערכים left\right לחילופין, כאשר right מפנה

ימינה ו-left שמאלה במידה ועשינו שמוש בשדה dir נשנה את ערכו.

במקרה הגרוע נכתב לעיל

גרסה ג' Tree\_b) (: בכל צומת קיים שדה בשם elem שמפנה לרשימה מקושרת. כשנכנס צומת חדש עם מפתח חוזר

הוא יוכנס לראש הרשימה.

במקרה הגרוע  *העץ יהיה מורכב ממפתחות זהים בלבד לכן כל ההכנסה של צומת חדש לעץ לוקחת (הכנסה*

*לראש הרשימה) \* n איברים סה"כ*

גרסה ד' ( :(Tree\_cהיא גרסה רנדומלית כאשר נכנס צומת בעל מפתח זהה פונים שמאלה או ימינה בצורה אקראית.

בגרסה זו הפניית מפתח זהה מתבצעת בצורה אקראית ולכן נחלק ל-2 מקרים:

* במקרה שבחירת הכיוון תהיה לצד אחד בלבד (ימין או שמאל) נקבל צורת עץ זהה לגרסה הראשונה (כמו רשימה

מקושרת) ולכן סיבוכיות זמן הריצה תהיה ג"כ זהה , זהו המקרה הגרוע.

* במקרה שבחירת הכיוון נעשתה פעם לימין ופעם לשמאל באופן קבוע נקבל עץ מאוזן זהה לעץ מגרסה שניה

ולכן סיבוכיות הזמן תהיה

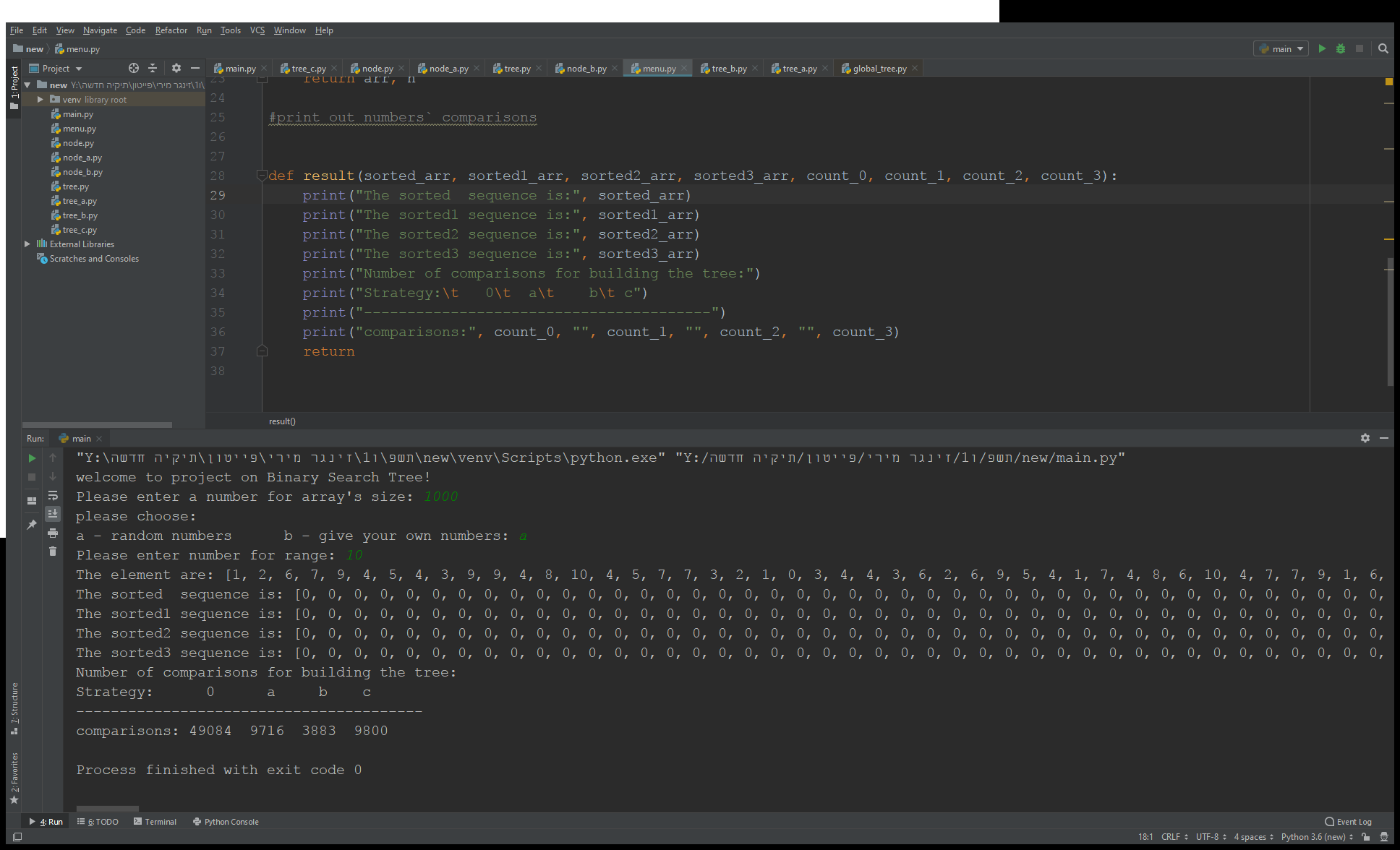
במקרה הממוצע גובה העץ יהיה ולכן בהכנסת n איברים נקבל

סעיף ב':

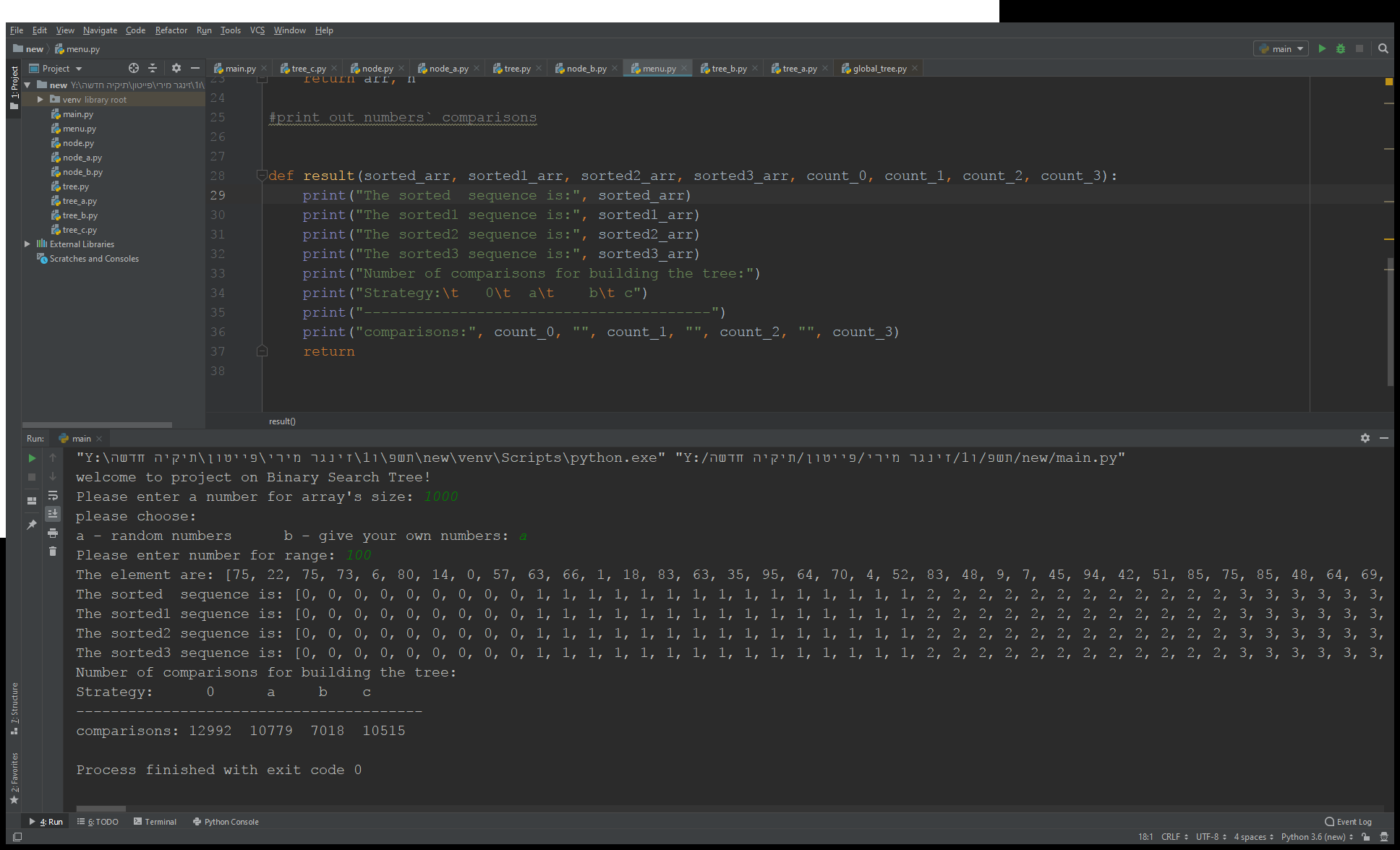
מצורפים קבצים המכילים את קוד הפרויקט.

סעיף ג':

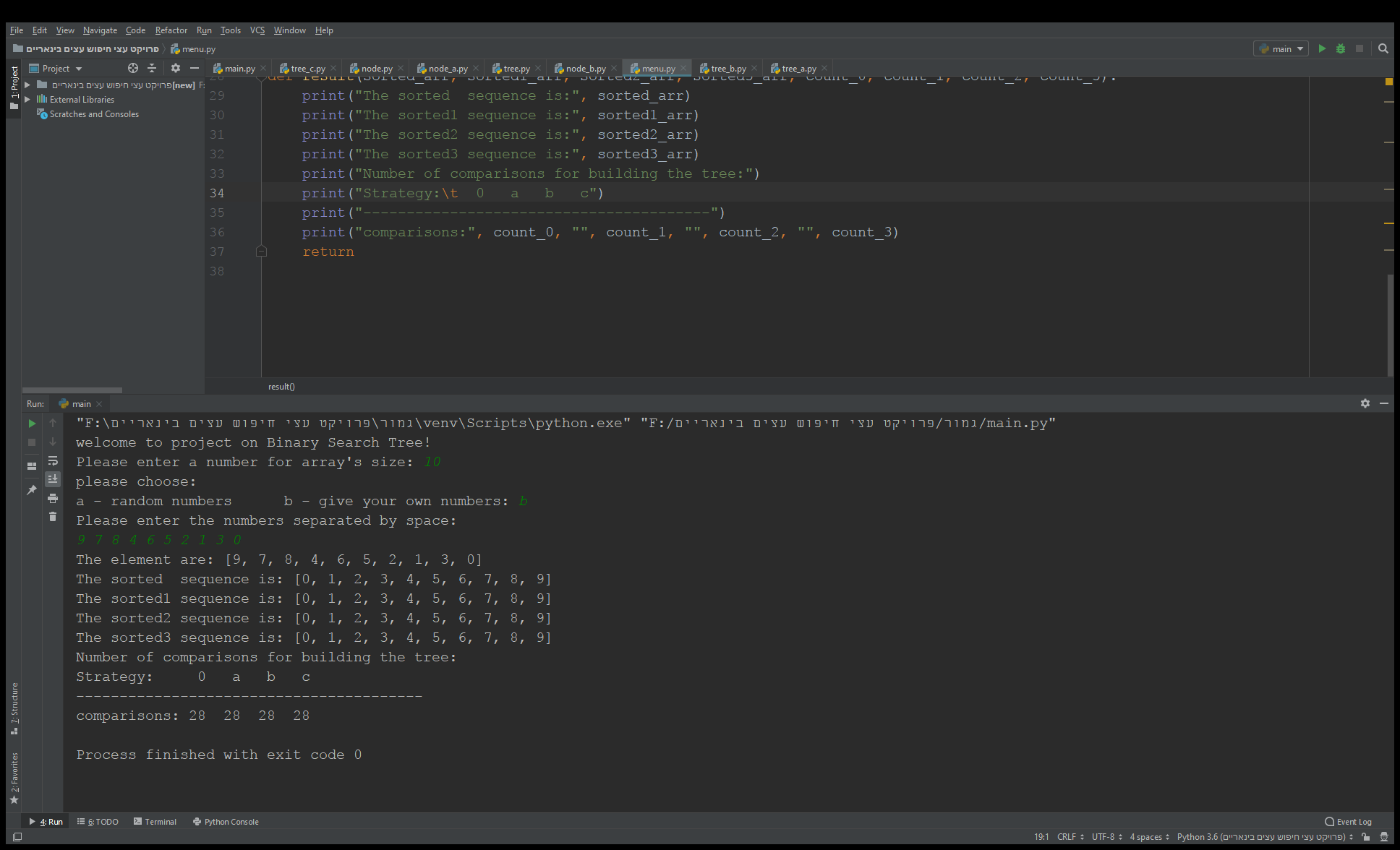
מס' ההשוואות לקלט אקראי באורך 1000 ובטווחים 10 ו- 100

דוגמאות הרצה:

עבור n=1000,range=10



עבור n=1000,range=100



עבור n=10 כאשר כל האיברים שונים זה מזה

סעיף ד':

בסעיף זה ננתח את התוצאות שקבלנו ע"י הרצת נתונים שונים-

גרסה א'-

עפ"י הדיאגרמה בסעיף קודם רואים שכמות ההשוואות הגבוהה ביותר הייתה בגרסה א' ואכן כשנתחנו את זמן הריצה בסעיף א'

קבלנו את זמן הריצה הגדול ביותר, ז"א העץ עורך כמות השוואות גדולה מאוד, בפרט כשטווח המספרים היה קטן מספר הצמתים

בעלי מפתחות זהים גדל באופן משמעותי ומספר ההשוואות גם הוא היה גדול במיוחד כי רב הפניות היו ימינה מה שגרם לכך

שהעץ יהיה עמוק יותר. ולכן גרסה זו אינה מומלצת עבור פתרון של בעיית עצי חיפוש בינאריים בעלי מפתחות זהים.

גרסהב'-

גרסה זו הינה גרסה משופרת יותר כיוון שגרסה זו יוצרת עץ מאוזן מפני שהכנסת האיברים מתבצעת ע"י השדה dir שמשתנה

לחילופין ימין/שמאל כך שכל רמה בעץ מתמלאת לגמרי לפני שיורדים לרמה נוספת.

וכן לא היה שינוי ממשי בטווח מספרים גדול או קטן. כמו שראינו בדיאגרמה את מספר ההשוואות בעת הכנסה לעץ שהיה כמעט

שווה.

גרסה ג'-

כפי שנכתב בתקציר הגרסה האופטימאלית היא גרסה ג'. כל איבר מכיל בתוכו שדה נוסף בשם elem שמצביע לראש רשימה

מקושרת. בשעה שהכנסנו צמתים בעלי מפתחות זהים הם שורשרו לתחילת הרשימה.

משום כך גובה העץ לא הושפע (קבלנו מבנה של עץ בעל מפתחות שונים ללא חזרות).

לעומת שאר העצים, בגרסה זו כאשר טווח המספרים היה גדול גובה העץ גדל גם כן ולכן בעת הכנסת איבר לעץ מספר ההשוואות

גדל.

גרסה ד'-

גרסה זו דומה לגרסה ב' גם כאן נוצר עץ מאוזן אבל השינוי הוא בבחירת הכיוון שנעשה בצורה רנדומלית ואכן מספר ההשוואות

היה מקביל למספר ההשוואות בגרסה ב'.

סיכום-

* בעת הכנסת מפתחות שונים לעץ אין שינוי בין הגרסאות השונות
* בעת הכנסת מפתחות זהים: -הגרסה הגרועה ביותר הינה גרסה א' העץ הסטנדרטי

-הגרסה הטובה ביותר הינה גרסה ג'

- גרסאות ב' וד' חופפות מאד באופן ביצוען בממוצע ולכן מספר ההשוואות היה כמעט זהה.