**מטלת מנחה 16**

**מגישים**: דניאל מרקובסקי, 207934100  
 שמוליק כהן, 322646878

**תיאור הרעיון הכללי של הפרויקט**: המטלה שבבסיס הפרויקט היא כתיבת אלגוריתם מקוון שמקבל ערכים ממשיים, אנו צריכים לדאוג שבכל נקודה, ניתן יהיה לדאוג להכניס ערך בזמן של: ולהחזיר חציון בזמן של: .

**תיאור מבנה הנתונים שנבחר**: מבנה הנתונים שבחרנו הוא עץ אדום שחור סטנדרטי.  
כמו כן, החלטנו לתחזק משתנה חציון ומשתנה size ששומר את גודל העץ. הרעיון שלנו הוא שההכנסה תתבצע באופן סטנדרטי כפי שנכתב בספר, את אלגוריתם מציאת החציון כתבנו בעצמנו והגדרנו אותו להיות האיבר שהוא הערך התחתון של (size+1)/2 בגודלו בעץ.

**יתרונות וחסרונות**: בקורס למדנו מספר סוגים של מבני נתונים. נסקור את היתרונות של המבנה שבחרנו:

* ההכנסה מתבצעת ב
* ניתן לתחזק את median כיוון שניתן למצוא לכל איבר את הבא אחריו בגודלו בעץ ואת הבא לפניו בגודלו בעץ ב.
* פשוטה יחסית למימוש.

נסקור את החסרונות של המבנה שבחרנו:

* זמן הבנייה של עץ בגודל לוקח זמן רב יחסית - .

הערה: היינו יכולים לממש זאת באמצעות 2 ערימות(מינימום ומקסימום) כפי נלמד בקורס, אך לדעתנו מימוש תוך כדי שימוש בעץ אדום שחור הוא פשוט יותר וברור יותר.

**אלגוריתמים מרכזיים**:

* אלגוריתם שמכניס ערך לעץ ב ומשתמש ב על מנת לדאוג שההכנסה תהיה כעץ אדום שחור. ממומש בדיוק כמו בספר חוץ מזה שיש הוספה של 1 לsize כל הכנסה ושיש קריאה ל fix\_Median , פעולות אלה לא משנות את הסיבוכיות.
* פונקציה שמחזירה את החציון הנוכחי ב. ראוי לציין שאחרי כל הכנסה אנחנו מבצעים תיקון למצביע החציון שנשמר על פי הערך שנכנס ומספר האיברים הקיימים.

פונקציה שמקבלת את הערך של הצומת האחרונה שנכנסה למערך ומשנה את המצביע של median אם צריך. כאשר נכנס ערך חדש לעץ יש 4 אפשרויות עבור מיקום החציון:  
1.גודל העץ החדש הוא זוגי(2k) והערך שנכנס קטן מmedian הנוכחי- במקרה כזה,median היה האיבר הk בגודלו בעץ ועכשיו הוא צריך להישאר k ולכן median צריך להיות האיבר הקודם בגודלו בעץ.  
2.גודל העץ החדש הוא זוגי(2k) והערך שנכנס גדול מmedian הנוכחי- במקרה כזה,median היה האיבר הk בגודלו בעץ ועכשיו הוא צריך להישאר k ולכן אין צורך לשנותו כיוון שהוספנו ערך גדול ממנו ולכן הוא עדיין הk בגודלו בעץ.

3.גודל העץ החדש הוא אי-זוגי(2k+1) והערך שנכנס קטן מmedian הנוכחי- במקרה כזה,median היה האיבר הk בגודלו בעץ ועכשיו הוא צריך להיות הk+1 ולכן אין צורך לשנותו כיוון שהוספנו ערך קטן ממנו והוא נהיה הk+1 בגודלו.

4.גודל העץ החדש הוא אי-זוגי(2k+1) והערך שנכנס גדול מmedian הנוכחי- במקרה כזה,median היה האיבר הk בגודלו בעץ ועכשיו הוא צריך להיות הk+1 ולכן median צריך להיות האיבר הבא בגודלו בעץ.

בהתאם ל4 מקרים אלה, כתבנו את הפונקציה .

**ניתוח סיבוכיות של הפעולות הנדרשות**: בכל האלגוריתמים המרכזיים, ציטטנו את העמודים בספר בהם מופיע האלגוריתם וניתוח הסיבוכיות שלו.  *עבור* fix\_Median *שהוספנו*, *הסיבוכיות* *היא* (O*(*log *n כיוון שאו שלא עשינו כלום או שהשתמשנו ב*tree\_Successor *או* tree\_Predecessor *והסיבוכיות של שתיהן היא* (O(log n.

*דוגמא לפלט(כאשר כל הקריאות לפונקציות* find\_median\_in\_the\_hard\_way *במיין הן לא בתור הערות):*

*first array: A, size: 200*

*median of the n/4 first numbers 460.43958152222376*

*median of the n/2 first numbers 511.7958479608704*

*median of the 3n/4 first numbers 536.9729817790044*

*460.43958152222376*

*511.7958479608704*

*536.9729817790044*

*second array: B, size: 400*

*median of the n/4 first numbers 502.7317522066783*

*median of the n/2 first numbers 553.6600111538065*

*median of the 3n/4 first numbers 540.8029411121365*

*502.7317522066783*

*553.6600111538065*

*540.8029411121365*

*third array: C, size: 800*

*median of the n/4 first numbers 471.8511257735341*

*median of the n/2 first numbers 507.04392594475223*

*median of the 3n/4 first numbers 508.66412401582437*

*471.8511257735341*

*507.04392594475223*

*508.66412401582437*