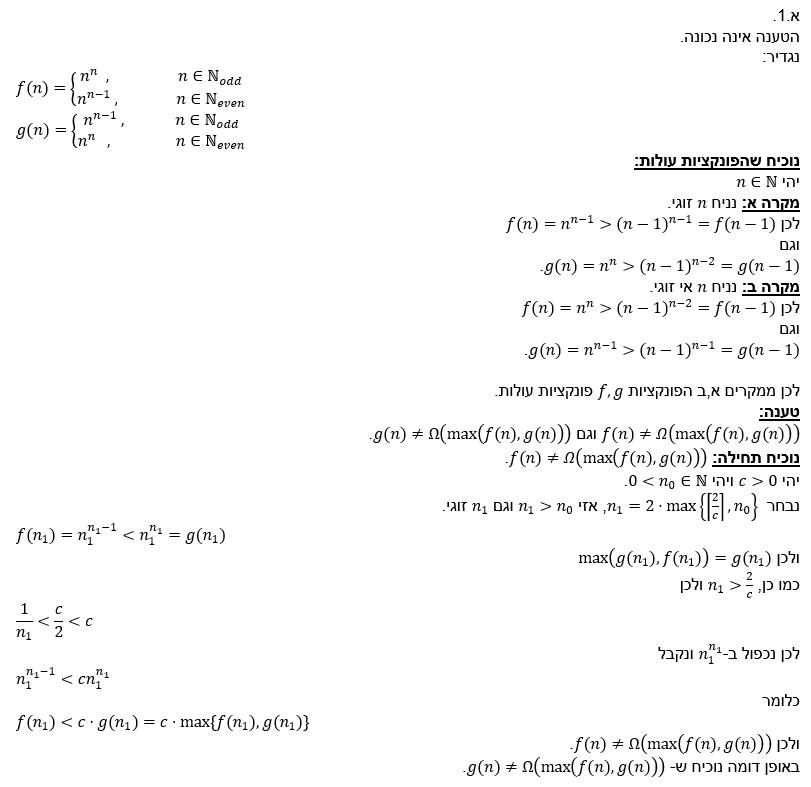
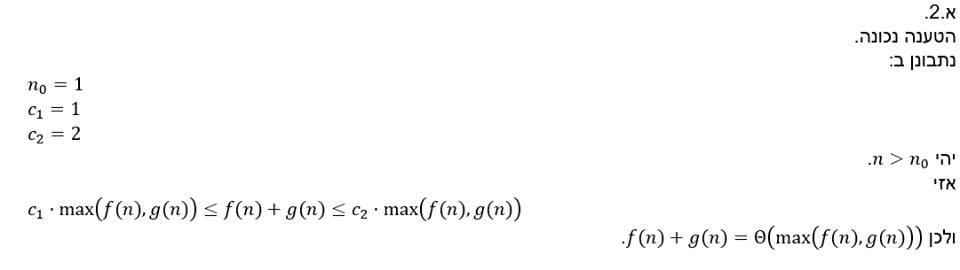
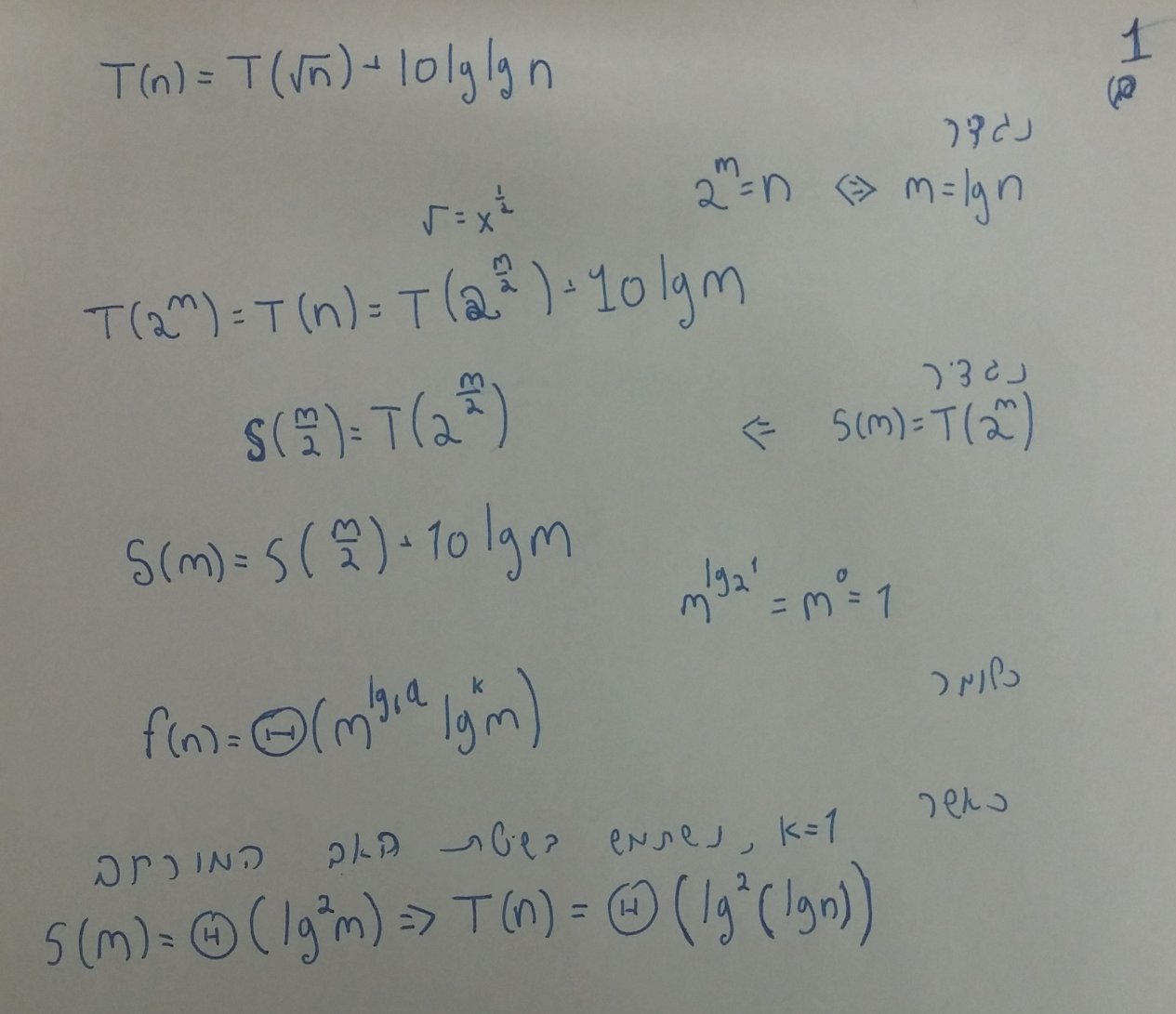
שאלה 1





ב-



שאלה 2

מכיוון שהמערך ממוין, נוכל לרוץ על העץ בסריקה תוכית יחד עם המערך ולהשוואת את האיברים.

נבצע שינוי בסריקה, במקום הדפסה, נבצע השוואה.

כאשר הסריקה תתחיל בהשוואות, היא תתחיל מאיבר המינימלי בעץ (מתוכנות הסריקה), ונעבור עם הסריקה על m אברי המערך, במידה וכל האיברים נמצאים, כאשר נגיע לאיבר ה m, ונגלה שהוא נמצא נצא מפעולת הסריקה.

נתחיל עם מצביע לתחילת המערך ונבצע את פעולת הסריקה.

כאשר נגיע לפעולת ההשוואה יכולים להתרחש מספר מיקרים.

* האיבר בעץ קטן מאיבר במערך, במקרה כזה רק הסריקה תתקדם להשוואה הבאה.
* האיבר שווה, המצביע של המערך יעבור לאיבר הבא והסריקה תתקדם להשוואה הבאה.
* האיבר שמחזירה השגירה גדול מאיבר בעץ, מכיוון שלא יצאנו מהשגירה סימן שאיבר לא נמצא במערך, ולכן נחזיר הודעת שגיאה.

לא היה שימוש במערכים נוספים, לכן נשארו בסיבוכיות מקום קבוע.

סיבוכיות זמן ריצה:

מעבר על המערך .O(m)

סריקה תוכית, ע"מ 214 בספר, .O(n)

מכיוון ש m<nלכן סה"כ זמן הריצה היה Θ(n).

שאלה 3

מבנה הנותנים היה עץ אדום שחור עם 2 שדות הרחבה.

size – כמות הבנים הימנים של הצמות ביחס לאותה הכנסה.

Inbefore – ערך האיברים הגדולים והוותיקים מהצומת.

תיאור האלגוריתם:

ברגע שנכניס איבר חדש לתוך העץ, נספור כמה איברים גדולים מאיבר שהוכנס.

הנכונות של הרעיון הזה בא מקח שבמבנה הנותנים הזה אין פעולה של מחיקה לכן כל איבר שקיים כבר בתוך המבנה חייב להיות ותיק יותר מהאיבר שמוכנס.

Insert – בעת הכנסה לעץ, במידה ואנו יורדים לעבר הבן הימני, השדה size שעברנו יגדל ב-1, אך לא נסכום אותו, כאשר יורדים לעבר הבן השמאלי סוכמים את ערך ה size של הצומת שעברנו ונוסיף לסכימה הזאת 1, במידה ושדה ההרחבה size של הבן השמאלי שלו קיים נחסר אותו מהסכימה.

כאשר האיבר הגיע למקומו ערך שדה ה Inbefore יקבל לערך הסכימה שלנו.

insert – חסם תחתון O(1) כלומר הכנסת השורש, חסם עליון O(logn) לכן סה"כ היה Θ(logn).

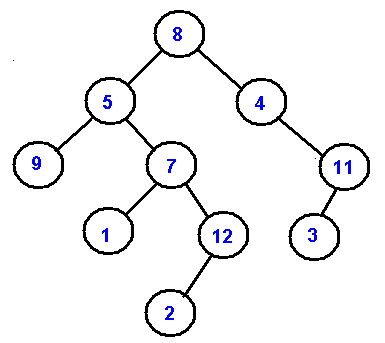
find – חסם תחתון O(1) כלומר מציאת השורש, חסם עליון O(logn) לכן סה"כ היה Θ(logn).

inversions- חיפוש רגיל, במידה ונמצא נחזיר את הערך Inbefore, מאותה סיבות כמו find סה"כ זמן הריצה היה Θ(logn).

בעת פעולת הסיבוב הרגילות, לא תהיה פגיע במבנה, כי השדה size יעבור בהתאם ובעת ההכנסה הבא עדין ישמר מה מספר האיברים הגדולים מאותה צומת ולכן ידע כמה איברים ותיקים יותר קיימים מאותה הכנסה.

שאלה 4

א – דוגמה לעץ ול2 המערכים

A-PostOrder - 9, 1, 2, 12, 7, 5, 3, 11, 4, 8

B-InOrder - 9, 5, 1, 7, 2, 12, 8, 4, 3, 11 

מתכונות סריקה סופית,POSTORDER,נובע שתמיד השורש של העץ האיבר האחרון של המערך, במקרה זה A[n], הוא 8, נגדיר אותו כשורש.

כעט נחפש את איבר השורש,8 במקרה זה, במערך של הסריקה התוכית, InOrder, כאשר מצאנו באיזה תא i הוא נמצא, נדע שכל האיברים שמתחת לi הם בצדו השמאלי וכל האיברים שמעל i הם בצדו הימני, במערך B כמובן.

כעת נגדיר X, כמספר האיברים מB[1] ל B[i],{ 9, 5, 1, 7, 2, 12} .

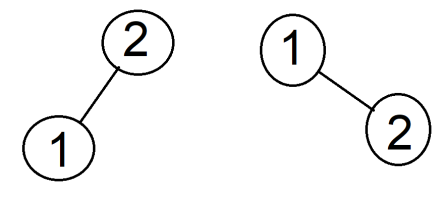
ונגדיר Y, כמספר האיברים מB[i] ל B[n],{ 4, 3, 11}.

כעת באופן דומה, נבנה את תת עץ שמאלי של העץ, כי אם נרוץ עם X, על A, נראה ש A[X]=5, הוא "השורש" של בתת עץ השמאלי וכך נרוץ עד סיום כל תתי העץ.

באופן זהה, נבע את התת עץ הימני עם Y.

.

סה"כ זמן ריצה Θ(n²)

ב – לא, לדוגמה 2 עצים שונים שיתנו תוצאת סריקה זהה.

שנהיים יתנו 1,2

שאלה 5

נשתמש בעץ אדום שחור מורחב שמכיל את מספר האיברים בתת העץ המושרש של אותה הצמות.

Insert- בדומה לצורה הרגילה, פרט לתוספת הבאה, בעת ההכנסה כל צומת שעוברים מגדלים את שדה ההרחבה של אותה צומת ב1.

חסם תחתון O(1) כלומר הכנסת השורש, חסם עליון O(logn) לכן סה"כ היה Θ(logn).

find – בדומה לצורה הרגילה ,חסם תחתון O(1) כלומר מציאת השורש, חסם עליון O(logn) לכן סה"כ היה Θ(logn).

Range- נעזר בשדה ההרחבה כדי למצאו את המפתח X, כאשר מתחיל לחפש מהשורש ועובר כל צומת נבדוק:

- אם המפתח בצומת הנוכחי קטן מX נסכום למשתנה זמני את הסכום שיש בשדה ההרחבה פחות הסכום שיש לבן הימני של הצומת ונתקדם ימינה.

- אם המפתח בצומת הנוכחי גדול שווה לX נתקדם שמאלה בלי להוסיף דבר לסכום שצברנו.

במידה הצומת שווה ממש לX נוסיף לסכום 1.

בסוף החיפוש נשאר עם סכום האיברים הקטנים ממש מX, מכיוון ששדה ההרחבה שלנו מצין כמה איברים יש בתת צומת, ולכן ברגע שמתקדים ימינה נובע שX, גדול יותר מהצומת שעברנו ולכן ניסכום כמה איברים קטנים מימנה קיימים.

נבצע חיפוש דומה על Y, נחפש בעץ ונסכום במשתנה זמני אחר את סכום האיברים הקטנים שווים לו. במידה ונתקלנו בצומת אשר שווה ממש לY נוסיף לסכום 1.

לבסוף נותרנו עם 2 משתנים זמנים SUMX וSUMY נחש את ההפרש בינהים ונקבל את סכום האיברים הקטנים שווים מY וגדולים שווים לX.

זמן הריצה, 2 חיפושים רגילים.

O(logn)+ O(logn)= Θ(logn).

תמיכה בסיבוב:

* סכומו של Y משתה לסכומו של Y שהיה פחות הסכום של ϒ פחות 2 (צמתים X ו- ϒ עצמם).
* סכומו של X משתנה לסכומו של Y אחרי הסיבוב פלוס הסכום של ϒ פלוס 2 (בצמתים ϒ ו-Y עצמם).

