

1. לא נכון, דוגמה בעמוד 42 ובעמוד 50 בספר, או לחילופין 2 פונקציות מתחלפות (זוגי ואי זוגי) כמו בממ"ן 11 2018א.
2. לא נכון, מיון מהיר במקרה הגרוע בו מתבצעת חלוקה גרוע יהיה , שזה לא אופטימלי. ספר עמוד 125. מסקנה 8.2 עמוד 139 – מיון ערימה ומיון-מיזוג הם מיוני השוואה אופטימליים אסימפטוטית.
3. לא נכון, עבור כל מיון מבוסס השוואות, החסם התחתון היינו – עמוד 139 בספר.

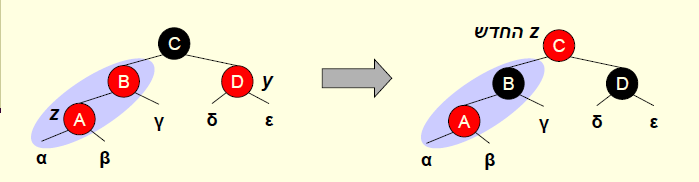
הנחה: לא נתון שום דבר על מספר הספרות, ולכן הוא יכול להיות גדול מ-n ולכן לא נוכל להשתמש במיון בסיס (עמוד 142). כמו כן לא נתון כי המספרים בתחום 0 עד K כלשהו, ולכן גם מיון מנייה לא מתאים כאן (עמוד 140 בספר).

1. לא נכון, לפי עמוד 117 בספר, זמן הריצה של max-heap-insert הוא O(logn). הכנסת איבר לערימה עלולה לגרום להפרה ולכן עלינו להפעיל השגרה heap-increase-keyשתמקם את האיבר במקום הנכון.
2. נכון, לפי עמוד 195 בספר, אם משתמשים במספר הזהב יש סיכוי סביר שהגיבוב יפעל בצורה טובה למדי.
3. לא נכון, סריקה של עץ חיפוש בינארי בעל n צמתים מתבצעת ב-, ללא תלות בגובה העץ. ספר עמוד 214.
4. נכון, ניתן להחזיק משתנה שמחזיק את מספר הצמתים בעץ, ובזמן הכנסה או מחיקה לבצע OS-SELECT(n/2) על מנת לעדכן את החציון.
5. לא נכון, דוגמה נגדית מתוך סריקה תוכית [1,2], מתקבלים שתי עצים שונים.
6. נכון, בהנחה שהתחום ידוע מראש אפשר לממש באמצעות מיעון ישיר, אך בזבזני מאוד מבחינת זכירון.
7. נכון, רוטציה ימינה עלולה גרום להפרה של התכונות 2,4,5. עמוד 230 בספר.

הסבר על רוטציות:

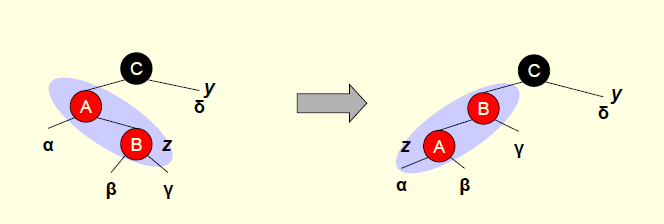
מקרה 1:

האבא והדוד אדומים, נבצע אותם בשחור ואת הסבא באדום, ונמשיך כך כל עוד יש הפרה עם הסבא.



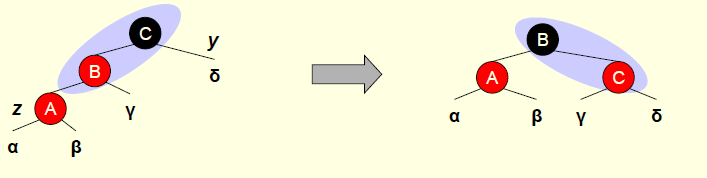
מקרה 2:

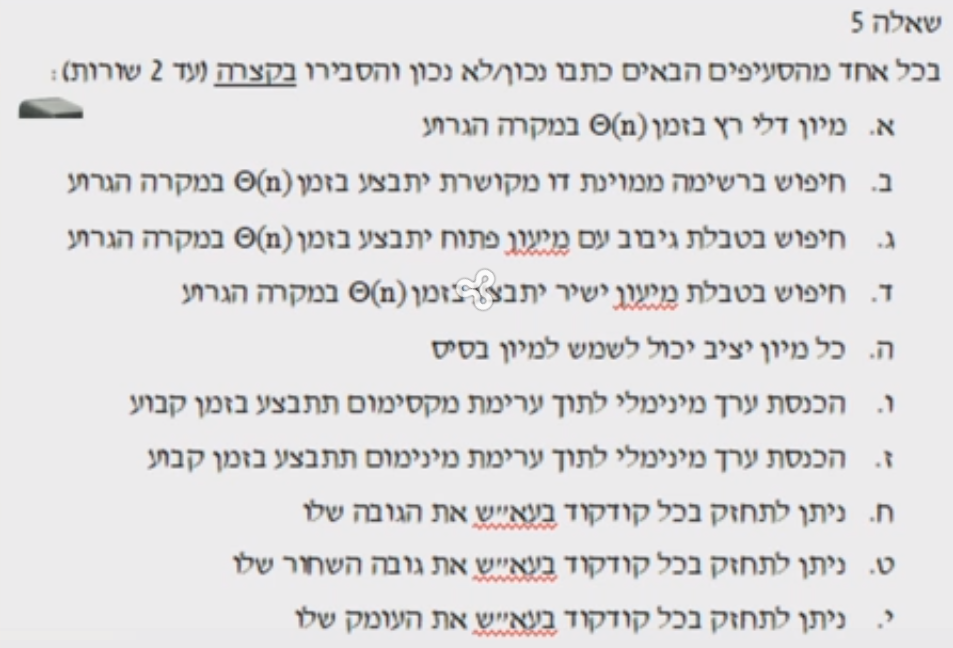
בן ימני של אבא אדום ודוד שחור, נבצע סיבוב שמאלי סביב האב האדום, ומעבור למקרה 3.



מקרה 3:

בן שמאלי של אבא אדום ודוד שחור. נבצע סיבוב ימני סביב האבא האדום וניצבע את האבא בשחור ואת הסבא באדום.





1. לא נכון, יתכן שכל האיברים ייפלו לאותו דלי ולכן זמן הריצה בדלי הספציפי הזה הוא לא לינארי.
2. נכון, מאחר ולא ניתן לקפוץ למקומות שונים ברשימה עדיין נצתרך לעבור באופן סידרתי עד למציאת האיבר.
3. נכון, במקרה נצתרך לבצע חיפוש בכל תאי הטבלה כגודל הטבלה ולכן המקרה הגרוע יהיה
4. לא נכון, לפי המפתח ישר ניגשים לתא בטבלה אין צורך לחפש אותו. ספר עמוד 187.
5. נכון, מיון בסיס דורש מיון יציב כדי לשמור על המיון בכל שלב, ולכן ניתן לבחור כל מיון יציב, אך הבחירה של המיון עלולה להשפיע על זמן הריצה. מיון מנייה הוא המיון האופטימלי עבור מיון בסיס. **מיו יציב:** מיון בועות, מיון הכנסה, מיון מיזוג, מיון מנייה, מיון בסיס

**מיון לא יציב:** מיון בחירה, מיון מהיר, מיון ערימה.

1. נכון, הכנסת איבר לערימה מתבצע ע"י הוספתו בתור עלה ולכן תכונת המקסימום תישמר. כל רמה בערימת מקסימום גדולה מהרמה מתחת לכן בעלים יהיו כל האיברים הקטנים יותר.
2. לא נכון, בעירמת מינימום השורש הוא האיבר המינימלי, ולכן אם נכניס איבר מינימום בתור עלה תהיה הפרה ונצתרך לעלות את הצומת החדש עד השורש של העץ.
3. נכון לפי משפט 14.1 – גובה של צומת הוא המסלול הארוך ביותר מצומת מסויים לעלה. ניתן לבחור את המקסימום מבין שת הילדים כדי לקבל גובה של צומת מסויים.
4. נכון לפי משפט 14.1 – אפשר לדעת את גובה השחור של צומת ע"י גובה השחור של הילדים של אותו הצומת ע"י הפרדה למקרים (האם הבן אדום או שחור נוסיף בהתאמה).
5. לא נכון, לא ניתן לחשב זאת באמצעות הילדים, ולכן משפט 14.1 לא מתקיים. לא ניתן לתחזק את המתרחש למעלה בעץ (מעל צומת מסויים).

**שאלות מהשיעור של אורי:**

1. קיימים קלטים שעליהם מיון הכנסה מהיר יותר ממיון מיזוג אסימפטוטית.

**נכון,** עבור מערך ממויין, מיון הכנסה עובד בסיבוכיות ומיון מיגוז ב-

1. לכל שתי פונקציות חיוביות f,g מתקיים:

**לא נכון:**

נניח ש- f=O(g)

אזי קיימים

כל שלכל מתקיים

בפרט עבור אי זוגי מתקיים

אבל אם ניקח

אז יתקיים ש- n אי זוגי ו- n>c – וזו סתירה.

לכן בדומה .

1. איתור מקסימום בערימת מינימום דורשת סיבוכיות במקרה הגרוע.

**נכון,** בערימת מינימום המקסימום נמצא באחת העלים בערימה ויש עלים.

אין שום ידיע על יחס סדר בין שני עלים בערימה. יש למצוא מקסימום בקבוצה כללית בגודל , וזה דורש סיבוכיות .

1. עבור 2 צמתים בעץ בינארי בו יש הצבעות גם מצמתים לאבותיהם, ניתן למצוא אב קדמון משותף נמוך ביותר בסיבוכיות .

**נכון**, אם הצמתים באותו עומק, ניתן לעלות שלב אחד בעץ בכל אחד מהם בו זמנית עד שמצביעים לאותו צומת. וזהו אב משותף קדמון הכי נמוך שלהם - .

אם הם בעומק שונה, צריך לחשב עומק של כל אחד משתי הצמתים –

ואם העומקים שונים, עולים בצומת הנמוך ביותר מספר שלבים זהה להפרש העומקים -

וממשיכים כמו קודם.

