פתרון לתרגיל 15.1-6 מהספר

נתון לנו עץ אדום-שחור, שבו יש בכל צומת שדה נוסף, המכיל את מיקומו של הצומת בתת-עץ המושרש בצומת. (כלומר, מיקומו של הצומת בסדר הלינארי הנקבע ע״י סריקה תוכית של צמתי התת-עץ.)

נקרא לשדה זה rank. עלינו להראות שניתן לעדכן את השדה rank בכל הכנסה או מחיקה.

הכנסה: כפי שמוסבר בספר, הכנסה לעץ אדום-שחור מתבצעת בשני שלבים -
הכנסה רגילה לעץ חיפוש בינרי ואח״כ ״תיקון״ העץ כדי לשמור על תכונות האדום-שחור.

כדי לעדכן את השדה rank בשלב הראשון, פשוט מוסיפים 1 ל-rank[x] בכל צומת x שממנו פונים **שמאלה** במהלך סריקת העץ מן השורש כלפי מטה.

השדה rank של הצומת החדש יקבל את הערך 1.

בשלב השני מתבצעות לכל היותר שתי רוטציות.

נשים לב, שבעת ביצוע רוטציה שמאלית, השדה rank של הצומת שעליו מופעלת הרוטציה אינו משתנה. לכן מספיק להוסיף שורה אחת לקוד של השגרה LEFT_ROTATE (T, x) :

13 rank[y] $\leftarrow$ rank[y] + rank[x]

השורה שצריך להוסיף לקוד של השגרה RIGHT_ROTATE (T, y) :

13 rank[y] $\leftarrow$ rank[y] – rank[x]

מחיקה: מחיקה מעץ אדום-שחור מתבצעת גם-כן בשני שלבים - הסרת הצומת מהעץ ואח״כ ״תיקון״ העץ כדי לשמור על תכונות האדום-שחור.

נסמן את הצומת המוסר מהעץ בשלב הראשון ב-y. כדי לעדכן את השדה rank, עלינו לעלות מ-y במעלה העץ אל השורש, ולהחסיר 1 מהשדה rank של כל צומת ש-y הוא בן **שמאלי** שלו.

בשלב השני מתבצעות לכל היותר שלוש רוטציות, ויש לשנות את הקוד של השגרות המתאימות כפי שהוסבר לעיל.

מכיוון שגובהו של עץ אדום-שחור הוא $O(\lg n)$, ברור שהעלות הנוספת הכרוכה בעדכון השדות rank היא $O(\lg n)$.