תיאור הבעיה

עץ חיפוש בינארי רגיל בן n צמתים מכיל n + 1 מצביעי Left ו-Right שערכם NIL. במצב זה זיכרון רב "מתבזבז".

הגישה הכללית של התוכנית לפתרון הבעיה

נחליף את מצביעי ה-Left המכילים NIL של הצומת z במצביע ל-.

באופן דומה, נחליף את מצביעי ה-Right המכילים NIL של הצומת z במצביע ל-TREE-SUCCESSOR(z).

כך נבנה עץ מחוות.

מבנה הנתונים

התוכנית מתבססת על עץ חיפוש בינארי, אליו *הוספנו את לוגיקת החוטים.*

*החוטים מתוחזקים בעת הכנסה ומחיקה מהעץ.*

*על מנת להבדיל בין מצביעים חוטים למצביעים לבנים אמיתיים, מימשנו 2 פונקציות עבור כל צומת – HasARightChild, HasALeftChild הבודקות האם לצומת יש בן אמיתי. במקרה שהמצביע הוא לא NIL, אנו בודקים אם האבא של הצומת הבן הוא הצומת הנוכחית, אם כן – זה הוא בן אמיתי.*