מגישים: רן לוטם, דביר פרי

תיאור מילולי וגרפי של מבנה הנתונים:

* המחלקה הראשית בשם MagicManager מכילה ארבעה מבני נתונים:
  + UnionFind של חיות: ממומש כמערך של חיות, המייצג עצים הפוכים. כל חיה מכילה מידע על הרמה שלה (בפועל, סכום הרמות של החיות באזור, אם היא השורש של האזור), מספר החיות באזור בו היא נמצאת, מצביע לבעלים שלה, אם יש כזה, ואת האינדקס במערך של החיה שהיא שורש האזור שלה.

המתודות אותן מממש מבנה הנתונים הן:

* + - אתחול מערך – יצירת מערך דינמי בגודל n (מספר החיות), ואתחול כל חיה עם הרמה שלה. סיבוכיות זמן היא לפיכך O(n).
    - מציאת חיה – הולכים לתא במערך שבו נמצאת החיה בעזרת ה-ID שלה. מטיילים במעלה העץ ההפוך (מהחיה, לחיה שמוגדרת כשורש שלה, וחוזר חלילה, עד שמגיעים לחיה שהיא השורש של עצמה) עד שמוצאים את החיה שהיא השורש של האיזור. לאחר מכן מבצעים כיווץ מסלולים, כלומר עבור כל החיות במסלול החיפוש, השורש מעודכן לשורש האמיתי. מכיוון שאנו מכווצים את המסלול של החיות שבדרך, ומכיוון שאנו נחבר את האיזור הקטן לאיזור הגדול (מוסבר באיחוד, בהמשך), מובטח לנו שמציאת חיה עורכת O(log\*n) משוערך.
    - הסרת מחסום בין חיות – ראשית, מוצאים את שתי החיות, מה שמבטיח ששורשיהם מעודכנים. עד כה O(log\*n) סיבוכיות זמן משוערכת. נבדוק איזה איזור גדול יותר, וכעת השורש של השורש של האיזור הקטן יותר, יהפוך להיות השורש של האיזור הגדול יותר. נעדכן את הגודל של השורש של האיזור הגדול, וגם את סכום הרמות של החיות בו. אלה פעולות הלוקחות O(1) זמן ולכן הסיבוכיות נותרת O(log\*n) משוערך.
    - החזרת מספר החיות באיזור – מוצאים את החיה בזמן O(log\*n) משוערך, ומחזירים את גודל האיזור ששמור בשורש של אותו איזור בזמן O(1) (גישה למערך). סה"כ O(log\*n) משוערך.
    - בדיקה האם שתי חיות באותו האיזור – מוצאים את שתי החיות בזמן O(log\*n) משוערך, ומחזירים האם לשתיהן אותה חיה בשורש. סה"כ O(log\*n) משוערך.