RAVL<Magi,Magi>availableMagiTree: זהו עץ דרגות אשר מכיל את המאגים מסודרים על פי הדרגה כאשר בכל צומת שמור מידע נוסף שהוא מצביע למאגי הכי צעיר בתת העץ ,בעץ זה נמצאים רק המאגים אשר לא אחראים על אף איזור

AVLTREE<int,Magi>allMagiTree : זהו עץ AVL אשר מכיל את כל המאגים שנמצאים במערכת (גם כאלו שאחראים על איזור ברגע זה וגם כאלו שאינם ) ,עץ זה מסודר לפי מספר הזהות של המאגים.

סיבוכיות פונקציות :

INIT : באתחול המערכת למעשה רק מאותחלים ארבעת מבני הנתונים לכן אתחול של עץ דרגות ריק בO(1) אתחול של עץ AVL ריק כנ"ל ב O(1) , אתחול של של UNIONFIND כמו שראינו בהרצאה בO(n) ואתחול טבלת ערבול ב O(1) לכן בסה"כ O(n) במקרה הגרוע.

AddMagiZoologist: בהוספת מאגי למערכת ,המאגי מוכנס לעץ allMagiTree ,כמו שלמדנו הכנסה לעץ AVL ב O(log(k)) כמו כן המאגי מוכנס לעץ availableMagiTree כנ"ל הכנסה לעץ ב O(log(k))(כמו שראינו תוך כדי ההכנסה לעץ אנחנו גם מעדכנים את המידע הנוסף השמור בכל צומת בלי לפגוע בסיבוכיות זו), לכן בסה"כ קיבלנו שבמקרה גרוע עושים זאת בO(log(k)) לכן כמובן שגם עומדים בO(log(k)) משוערך בממוצע על הקלט כמו שראינו בתרגול.

RemoveMagiZoologist: בהסרה מהמערכת למעשה יש חיפוש והסרה מטבלת ערבול (עם מערכים דינאמיים וdouble hashing ) אשר כמו שראינו בהרצאות נעשה בO(1) משוערך בממוצע על הקלט בנוסף יש הסרה משני עצים שכל אחד נעשה בO(log(k)) לכן בסה"כ נקבל O(log(k)) משוערך בממוצע על הקלט

RemoveBarrier : זוהי למעשה אך ורק פעולת JOIN של unionfind עם כיווץ מסלולים ואיחוד לפי גודל לכן כפי שלמדנו זה נעשה בO(log\*(n)) משוערך

AssignMagizoologistToCreature : בפעולה זו ראשית אנו מסתכלים בAnimalZoneUF ומחפשים את רמת הסיכון של האיזור מציאת האיזור היא למעשה פעולת FIND של unionfind עם כיווץ מסלולים ואיחוד לפי גודל לכן נעשית בO(log\*(n))