מבני נתונים - רטוב 1

מבני הנתונים שלנו מורכב ממבני הנתונים הפנימיים הבאים:

1. **מאגיזאולוג**

ייצג מאגיזאולוג אחד ,כל מאגיזאולוג יכיל שני עצים אשר יכילו מצביעים לכל היצורים שלו ,כך שאחד ממוין לפי מספר מזהה של היצורים ואחד ממוין לפי רמת סיכון. בנוסף כל מאגיזאולוג בעץ יכיל מצביע ליצור הכי מסוכן ששייך לו ואת מספר ה id שלו.

1. **יצור**

ייצג יצור אחד , יכיל מצביע למגיזאולוג שלו הוא שייך , מצביע לnode שלו בעץ שממויין לפי מספר מזהה אצל המאגיזאולוג ומצביע ל node שלו בעץ שממויין לפי רמת סיכון אצל המאגיזאולוג. כמו כן, יצור מכיל את רמת הסיכון שלו.

1. **עץ AVL של מאגיזאולוגים**

העץ יכיל בתוכו מצביעים לכל המאגיזאולוגים שנמצאים ממוינים לפי ה id שלהם כך שבהיתן id מסוים יהיה ניתן לגשת למאגיזאולוג ב O(logn)

1. **עץ AVL של יצורים ממוינים לפי מספר מזהה**

העץ יכיל בתוכו את כל היצורים במבנה הנתונים ממוינים לפי מספר מזהה כך שיהיה ניתן לגשת לכל יצור ב O(logn) .

1. **עץ AVL של יצורים ממוינים לפי רמת סיכון**

העץ יכיל בתוכו את כל היצורים במבנה הנתונים ממוינים לפי מספר מזהה כך שיהיה ניתן להדפיס את כל היצורים ממונים לפי רמת סיכון ב O(n).

פירוט הפונקציות של מבנה הנתונים:

1. Void\* Init()

מתבצע אתחול של 3 עצים ריקים ומכיוון שאתחול של עץ ריק לוקח סיבוכיות זמן ריצה של O(1) אז נקבל בסה"כ O(1) .

1. StatusType AddMagizoologist(void\* DS,int MagiID)

מתבצעת הכנסה של המאגיזולוג לעץ המאגיזולוגים ומכיוון שעץ הוא עץ AVL המכיל k מאגיזולוגים מתקיים כי ההכנסה נעשית בסיבוכיות זמן של O(1)

1. StatusType AddCreature(void\* DS,int CreatureID ,int MagiID,int Level)

מתבצעות הפעולות הבאות:

* מציאה של המאגיזולוג בעץ המאגיזולוגים שמכיל k מאגיזולוגים – O(logk)
* מכניסים את היצור לתוך עץ של המאגיזולוג הממוין לפי מספר מזהה – O(logn)
* מכניסים את היצור לתוך עץ של המאגיזולוג הממוין לפי רמת סיכון – O(logn)
* מכניסים את היצור לעץ כל היצורים הממוין לפי מספר מזהה – O(logn)
* מכניסים את היצור לעץ כל היצורים הממוין לפי רמת סיכון – O(logn)

בסה"כ נקבל

\*כאשר מחפשים בתוך העצים שנמצאים במאגיזאולוג הזמן הוא O(logn) מכיוון שבמקרה הכי גרוע כל היצורים שהוכנסו למבנה הנתונים הוכנסו אל המאגיזאולוג הזה.

1. StatusType ReleaseCreature(void\* DS,int CreatureID)

* נמצא את היצור העץ היצורים הממוין לפי מספר מזהה – O(logn)
* נשתמש במצביע שנמצא ביצור כדי להגיע למאגיזאולוג שלו – O(1)
* נסיר את היצור מעץ של המאגיזאולוג שממוין לפי מספר מזהה - O(logn)
* נסיר את היצור מעץ של המאגיזאולוג שממוין לפי רמת סיכון - O(logn)
* נסיר את היצור מעץ של כל היצורים שממוין לפי מספר מזהה - O(logn)
* נסיר את היצור מעץ של כל היצורים שממוין לפי רמת סיכון - O(logn)

בסה"כ נקבל

1. StatusType ReplaceMagizoologist(void\* DS,int MagiID,int ReplacementID)

מתבצעות הפעולות הבאות:

* מציאת המאגיזולוג הראשון בעץ המאגיזולוגים הממוין לפי מספר מזהה – O(logk)
* מציאת המאגיזולוג השני בעץ המאגיזולוגים הממוין לפי מספר מזהה – O(logk)
* סיור inorder על שני ועצים של המאגיזולוג הראשון וקבלת שני מערכים לכל עץ כאשר אחד מכיל את האינדקסים / רמות סיכון ממוינים ושני מכיל את היצורים שלהם בהתאמה (הפיכת כל עץ לשני מערכים) . במהלך הסיור עוברים על כל אחד מיצורים של המגיזאולוג פעם אחד ולכן הסיבוכיות זמן היא –O(nmagi)
* באותו אופן כמו בסעיף הקודם הופכים כל אחד מהעצים של המגיזאולוג לשני מערכים ולכן נקבל סיבוכיות זמן של O(nreplacement)
* נשתמש באלגוריתם merge של מיזוג שני מערכים ממוינים כדי למזג כל מערך של המגיזולוג הראשון עם הסימטרי שלו אצל מאגיזולוג שני, כך שהמיזוג של מערכי היצורים נעשה בהתאמה למיזוג מערכי המספרים המזהים/רמת סיכון.

יש 4 מיזוגים ולכן נקבל סיבוכיות זמן ריצה של

* נמחק את המגיזאולוג הראשון מעץ המאגיזאולוגים – O(logk)
* נמחק את שני העצים של המאגיזאולוג השני – 2\*O(nreplacement)
* ניצור שני עצים כמעט מלאים שיכולים להכיל כמות של יצורים -
* נעבור על כל אחד מן העצים בסיור Inorder ונמלא אותו לפי המערך המתאים לו
* נעדכן את היצור הכי מסוכן של המאגיזולוג השני להיות היצור שנמצא אחרון במערך ,נמצא אותו בעץ ונשמור מצביע ל node שלו אצל המאגיזאולוג - O(logk)
* עבור כל אחד מהיצורים של המאגיזאולוג השני נגדיר שהמגיזאולוג השני הוא המגיזאולוג שלהם

בסה"כ נקבל

1. StatusType IncreaseLevel(void\* DS,int CreatureID ,int LevelIncrease)

* נמצא את היצור בעץ של היצורים הממוין לפי מספר זהות – O(logn)
* נשתמש במצביע למגיזאולוג של היצור כדי למצוא את המאגיזאולוג – O(1)
* נמחק את היצור משני העצים של המאגיזאולוג –
* ביצור רשום מה היא רמת הסיכון שלו ואז בעזרת רמת הסיכון ומספר הזהות ניתן למצוא ולמחוק את היצור מעץ של היצורים הממוין לפי רמות סיכון וגם מהעץ הממוין לפי מספרי זהות– O(logn)\*2
* נוסיף את היצור עם הרמת סיכון המעודכנת לשני העצים של היצורים - O(logn)\*2
* נוסיף את היצור לשני העצים של המאגיזאולוג - O(logn)\*2 (כבר יש לנו מצביע אל המאגיזאולוג ולכן אין צורך לחפש אותו)
* אם היצור יותר מסוכן מהיצור הכי מסוכן הקודם אז נגדיר אותו להיות היצור הכי מסוכן החדש ונשמור מצביע אליו – O(1)

בסה"כ נקבל

1. StatusType GetMostDangerous(void\* DS, ,int MagiID, int CreatureID)

אם MagiID קטן מ 0 נחזיר את המספר המזהה של היצור הכי מסוכן שאותו שמרו ועדכנו מראש - O(1)

אם :

* נמצא את המגיזאולוג בעץ המאגיזאולוגים בעזרת המספר המזהה שלו – O(logk)
* בתוך המאגיזאולוג שמור המספר המזהה של היצור הכי מסוכן שלו ,ולכן נוכל להחזיר אותו ב O(1)

בסה"כ

1. StatusType GetAllCreaturesByLevel(void\* DS, ,int MagiID, int\*\* Creature, int\* numofCreatures)

אם :

* נבצע סיור inorder על העץ של כל היצורים שממוין לפי רמת סיכון ונשמור את האינדקס של כל יצור שעוברים עליו במערך – O(n)

אם :

* נמצא את המאגיזאולוג בעץ של המאגיזאולוגים – O(logk)
* נבצע סיור inorder על העץ של כל היצורים של המאגיזאולוג שממוין לפי רמת סיכון ונשמור את האינדקס של כל יצור שעוברים עליו במערך –

בסה"כ

1. Void Quit(void\*\* DS)

* נעבור בסיור postorder על העץ של המגיזאולוגים ונמחק אותו O(k)
* נעבור על עץ היצורים הממוינים לפי מספר זיהוי בסיור inorder ונמחק כל יצור שנעבור עליו – O(n)
* נעבור על עץ היצורים הממוינים לפי מספר רמת סיכון inorder ונמחק כל יצור שנעבור עליו – O(n)

בסה"כ נקבל

סיבוכיות מקום :

* עץ היצורים הממוינים לפי מספר רמת סיכון מכיל nיצורים – O(n)
* עץ היצורים הממוינים לפי מספר מזהה מכיל n יצורים – O(n)
* *עץ המאגיזאולוגים מכיל* k *מאגיזאולוגים ו* 2n *יצורים -*

*בסה"כ נקבל*