**מבני נתונים 1**

**גליון (רטוב) 1 (החלק היבש)**

**מגישים:**

|  |  |  |
| --- | --- | --- |
| שם | ת.ז | מייל |
| אילון הלוי | 328137831 | [eilon.halevy@campus.technion.ac.il](mailto:eilon.halevy@campus.technion.ac.il) |
|  |  |  |

**דרישות מבנה הנתונים:**

|  |  |  |
| --- | --- | --- |
|  | יצירת אובייקט חדש שמתאר רשימה ריקה. |  |
|  | מחיקת מבנה הנתונים (פינוי כל הזיכרון) |  |
|  | הפעולה מוסיפה ספינה חדשה למבנה הנתונים (במידה וספינה אם אותו מזהה לא קיימת) |  |
|  | הפעולה מסירה את הספינה בעלת המזהה  *מתוך מבנה הנתונים, זאת במידה וקיימת ספינה בעלת המזהה הנ"ל וגם ספינה זו ריקה מפיראטים.* |  |
|  | במידה ולא קיים פיראט בעל המזהה  *על הספינה, וכן קיימת ספינה בעלת המזהה מוסיפים את הפיראט לספינה.* |  |
|  | אם במבנה הנתונים קיים הפיראט בעל המזהה *, מוחקים אותו ממבנה הנתונים.* |  |
|  | במידה וקיימות ספינות במבנה הנתונים עם המזהים , *, ובנוסף הספינה בעלת המזהה*  אינה ריקה מפיראטים, מעבירים את הפיראט הותיק ביותר על הספינה בעלת המזהה  *לספינה בעלת המזהה .* |  |
|  | במידה וקיים פיראט בעל המזהה  *במבנה הנתונים, לשנות את ערך האוצר שלו ב-* *מטבעות.* |  |
|  | מחזיר את הערך של אוצר הפיראט בעל המזהה  *במידה וזה קיים במבנה הנתונים.* |  |
|  | במידה וקיימת ספינה בעלת המזהה  *במבנה הנתונים, מחזירים את מספר התותחים של הספינה.* |  |
|  | במידה וקיימת הספינה בעלת המזהה  *במבנה הנתונים, ובנוסף ספינה זו אינה ריקה מפיראטים, מחזירים את הפיראט העשיר ביותר בספינה.* |  |
|  | במידה וקיימות הספינות עם המזהים ,  *במבנה הנתונים, בודקים איזו ספינה מנצחת לפי כמות התותחים המאויישים, כל פיראט מהספינה המפסידה נותן לכל הפיראטים מספינה המנצחת מטבע מהאוצר שלו.* |  |

בנוסף, ממבנה הנתונים נדרש לעמוד בהגבלה של סיבוכיות זיכרון במבנה עצמו, ובכל הפעולות.

**מימוש:**

בחרנו לממש את מבנה הנתונים בעזרת עץ של ספינות שבו כל ספינה מחזיקה עץ של (מצביעים של) פיראטים ורשימה מקושרת דו-כיוונית של (מצביעים של) פיראטים.

את העץ הראשי (עץ הספינות) נמיין לפי ה- של הספינות. עבור כל ספינה עץ הפיראטים שלה ימוין על פי כמות המטבעות של כל פיראט (במקרה של כמות זהה של מטבעות נמיין לפי ה-), ואילו הרשימה המקושרת תמוין על פי סדר הכנסת לספינה (בדומה למבנה הנתונים- תור).

כמו כן, נחזיק בנפרד עץ של כלל הפיראטים הממוין לפי ה- כדי לתמוך בגישה לפיראטים לפי המזהה.

נוחות מימוש:

שדות הפיראט:

* מזהה הפיראט
* מצביע לספינה עליה הוא נמצא
* המצביע ל- של הרשימה הדו-כיוונית של הספינה בה נשמר המצביע לפיראט זה
* גודל האוצר של אותו פיראט, ללא החלק שהוא מפקיד בספינה

שדות הספינה:

* מזהה הספינה
* מספר התותחים של הספינה
* המאזן הכלכלי של הספינה (כל הפיראטים מפקידים בספינה אות אותה כמות מטבעות)
* רשימה מקושרת דו-כיוונית של מצביעי הפיראטים שנמצאים על אותה הספינה
* עץ של מצביעי הפיראטים שנמצאים על אותה הספינה שממויינים לפי איזה פיראט עשיר יותר.  
  (פרטי מימוש- יצרנו אובייקט שמחזיק במצביע ועושה אובר-לואוד לאופרטורים )
* מצביע לפיראט העשיר ביותר

הערה: כל פיראט יימצא על ספינה אחת בלבד! תמיד (עד שמוסר ממבנה הנתונים)

שדות האוקיינוס (מבנה הנתונים):

* העץ הראשי (עץ של ספינות, ממויין לפי )
* עץ כלל הפיראטים (ממויין לפי )

סיבוכיות הזיכרון של מבנה הנתונים:

* גודל העץ הראשי הוא כמספר הספינות
  + בכל ספינה, גודל עץ הפיראטים על אותה ספינה הוא מספר הפיראטים באותה ספינה
  + בכל ספינה, גודל הרשימה המקושרת הדו-כיוונית באותה ספינה
* גודל עץ כלל הפיראטים הוא כמספר הפיראטים הכולל במבנה הנתונים (באוקיינוס)

אנו יודעים שסיבוכיות המקום של רשימה מקושרת הן של עץ היא , כאשר זהו גודל מבנה הנתונים (מספר האיברים במבנה הנתונים). בנוסף, כל פיראט נמצא בספינה אחת בדיוק, ולכן סכום גדלי כל הרשימות המקושרות של כל הספינות, כמו סכום גדלי כל עצי של כל הספינות הוא מספר הפיראטים.

לכן, גודל מבנה הנתונים הכולל הוא כנדרש:

נוכיח כעת את סיבוכיות זמן הריצה של כל הפעולות

הפעולה *:*

*הוספה של איבר חדש לעץ היא בזמן ריצה , כאשר הוא גודל עץ (מספר הספינות).*

*הפעולה נכשלת אם קיים בעץ איבר עם אותו מזהה. כנדרש.*

*הפעולה* :

*הסרה של איבר מעץ היא בזמן ריצה , כאשר הוא גודל עץ (מספר הספינות).*

*הפעולה נכשלת אם לא קיים בעץ איבר עם אותו מזהה. כנדרש.*

*הפעולה* :

בדיקה האם קיים פיראט בעל אותו מזהה במבנה הנתונים- חיפוש בעץ הפיראטים, זוהי פעולה בזמן ריצה *, כאשר הוא גודל עץ (מספר הפיראטים).*

חיפוש הספינה בעלת מזהה במבנה הנתונים- חיפוש בעץ הראשי, זוהי פעולה בזמן ריצה *, כאשר הוא גודל עץ (מספר הספינות).*

אם הפעולות צלחו (לא נמצא פיראט ונמצאה הספינה), נוסיף את הפיראט לספינה:

* הכנסה לרשימה מקושרת (לתחילתה) ושמירת המצביע באובייקט הפיראט, .
* הכנסה של הפיראט לעץ הספינה . (גודל עץ הספינה חסום ע"י גודל עץ כלל הפיראטים)
* עדכון הפיראט העשיר ביותר (השורש השמאלי ביותר בעץ הספינה), זמן ריצה , חסם לגובה העץ.

סה"כ: , כנדרש.

הפעולה *:*

בדיקה האם קיים פיראט בעל אותו מזהה במבנה הנתונים- חיפוש בעץ הפיראטים, זוהי פעולה בזמן ריצה *, כאשר הוא גודל עץ (מספר הפיראטים).* אם נמצא הפיראט- נסיר את הפיראט מהספינה:

* גישה לספינה באמצעות המצביע שבשדה הפיראט, .
* הסרת ה- מהרשימה המקושרת הדו-כיוונית של אותה הספינה (באמצעות המצביע שבשדה הפיראט), .
* הסרת הפיראט מעץ הפראטים של הספינה, . (גודל עץ הספינה חסום ע"י גודל עץ כלל הפיראטים)
* עדכון הפיראט העשיר ביותר (השורש השמאלי ביותר בעץ הספינה), זמן ריצה , חסם לגובה העץ.
* לבסוף, נסיר את הפיראט מעץ כלל הפיראטים, .

סה"כ: , כנדרש.

הפעולה *:*