**החלק היבש בתרגיל בית רטוב 2 במבני נתונים**

**מגישים:**

**גיא אוחיון, ת"ז 315823856**

**רון קנטורוביץ', ת"ז**

תחילה נתאר את תכולת מבנה הנתונים והאובייקטים שאנו משתמשים בהם, ולאחר מכן נעבור לפירוט מעמיק לגבי המתודות על מנת להוכיח את הסיבוכיות שלהן. נתאר את מבנה הנתונים מהאובייקט הכי פנימי ועד המעטפת הכללית והחיצונית ביותר:

* Super Pixel: אובייקט שמייצג קבוצה אחת של פיקסלים, ומאחסן תיוגים וניקודים בעץ דרגות AVL (עבור כל תיוג יש ניקוד יחיד). התיוגים יהיו המפתחות בעץ, והניקוד יהיה הערך בכל צומת (כל ניקוד כמובן שייך למפתח ספציפי). הדרגה היא הניקוד, כלומר בשורש העץ קיים הניקוד הגבוהה ביותר, ולכן שליפה של הניקוד הגבוהה ביותר מהעץ מתבצעת בסיבוכיות זמן .



* Union-Find database: מאגר נתונים שמאחסן Super Pixels, תוך שימוש במבנה נתונים מסוג Union- Find שלמדנו בקורס. בנוסף, במאגר הנתונים אנו שומרים את כל הפיקסלים ששייכים לתמונה כלשהי, כך שכל פיקסל משויך ל-Super Pixel אחד ויחיד.



* Image: מעטפת פשוטה למאגר הנתונים של ה-Super Pixels. בכל תמונה יש קבוצות שונות של Super Pixels, ובנוסף מזהה תמונה.



* Hash Table דינמי מסוג Chain Hashing שלמדנו בקורס. מאחסן תמונות לפי המזהה של כל תמונה, תוך שימוש בפונקציית ערבול דינמית.



כעת נסביר כיצד עובדות כל המתודות במבנה הנתונים:

* : נאתחל משתנה מהמחלקה Hash Table שייצרנו, כך שגודל המערך יהיה 50. בכל אחד מהתאים במערך נבצע השמה לפוינטר . לבסוף, נחזיר מצביע למבנה הנתונים. אתחול מערך בגודל קבוע, וביצוע השמות אל מערך זה, מתבצעים **בסיבוכיות זמן ומקום במקרה הגרוע.**
* : נבדוק את תקינות הפרמטרים שהתקבלו, ונחזיר שגיאות מתאימות במידה ואחד מהן לא תקין בסיבוכיות זמן . אחרת, נבדוק את גודל ה-Hash Table הנמצא במבנה הנתונים . אם ה-Hash Table מלא, **נגדיל** את גודלו פי 2, נעדכן את פונקציית ה-Hash בהתאם, ונעדכן את כל התמונות שהיו בטבלה הקודמת לטבלה המתאימה לפונקציית ה-Hash החדשה (בדיוק כפי שלמדנו בתרגול – מערך דינמי של Hash Table). כעת, נחשב את האינדקס של התמונה המתאים ל-Hash Table על פי מזהה התמונה תוך שימוש בפונקציית ה-Hash המוגדרת במבנה הנתונים (נסמן את האינדקס ב-). נייצר תמונה בעלת מזהה , ובנוסף נבנה את מבנה הנתונים שמכיל את כל הסופר פיקסלים. מכיוון שיש לנו פיקסלים, סיבוכיות הזמן לבנות את המאגר תהיה (כי כל פיקסל הוא בפרט סופר פיקסל ריק בהתחלה). לאחר מכן, נכניס את התמונה לרשימה המקושרת הנמצאת באינדקס של מערך ה-Hash Table השייך למבנה הנתונים. ייצור תמונה, ביצוע כמות סופית של חישובים, והכנסת איבר לראש רשימה מקושרת מתבצעים בסיבוכיות זמן ומקום במקרה הגרוע. בנייה של מבנה בגודל מתבצעת בסיבוכיות זמן ומקום , כפי שלמדנו בקורס. מכיוון שאנו משתמשים ב, סיבוכיות הזמן המשוערכת עבור מחיקה והוספה של איברים ל-Hash Table היא . לכן, סיבוכיות הזמן המשוערכת של מתודה זו תושפע מ-, כמות הפיקסלים בתמונה. כלומר, סיבוכיות הזמן המשוערכת היא , כנדרש.
* : נבדוק את תקינות הפרמטרים שהתקבלו, ונחזיר שגיאות מתאימות במידה ואחד מהן לא תקין בסיבוכיות זמן . אחרת, נבדוק את גודל ה-Hash Table הנמצא במבנה הנתונים . אם ה-Hash Table מלא, **נקטין** את גודלו פי 2, נעדכן את פונקציית ה-Hash בהתאם, ונעדכן את כל התמונות שהיו בטבלה הקודמת לטבלה המתאימה לפונקציית ה-Hash החדשה (בדיוק כפי שלמדנו בתרגול – מערך דינמי של Hash Table). כעת, נחשב את האינדקס של התמונה המתאים ל-Hash Table על פי מזהה התמונה תוך שימוש בפונקציית ה-Hash המוגדרת במבנה הנתונים (נסמן את האינדקס ב-). אם התמונה לא קיימת, נחזיר ערך שגיאה. אחרת, נשלוף את התמונה מהרשימה המקושרת המתאימה לה בטבלת ה-Hash, ונמחק את התמונה. נתחיל בלמחוק את כל העצים בסופר פיקסלים השייכים לתמונה: בתמונה יש בסה"כ תיוגים, ולכן הגודל הכולל של כל העצים הנמצאים בסופר פיקסלים השייכים לתמונה הוא , ולכן סיבוכיות הזמן למחיקת כל העצים בכל הסופר פיקסלים היא . נמחק כעת את מבנה הנתונים שמאחסן את כל הסופר פיקסלים. התחלנו עם סופר פיקסלים ולכן יש לנו לכל היותר קבוצות, ולכן סיבוכיות הזמן למחיקה היא . נוציא את התמונה מהרשימה המקושרת שהיא נמצאת בה, נעדכן את הרשימה בהתאם, ואז נמחק את התמונה. ללא מחיקת העצים ומבנה הנתונים , סיבוכיות הזמן המשוערכת למחיקת תמונה היא , כפי שלמדנו בתרגול. לכן ,סיבוכיות הזמן המשוערכת של מתודה זו תהיה , כנדרש.
* : נבדוק את תקינות הפרמטרים שהתקבלו, ונחזיר שגיאות מתאימות במידה ואחד מהן לא תקין בסיבוכיות זמן . נחפש את התמונה בטבלת ה-Hash, ובמידה והיא לא נמצאת נחזיר שגיאה. אחרת, ניגש למבנה הנתונים הנמצא בתמונה, ששומר את קבוצות הסופר פיקסלים השייכים לתמונה. נמצא את הקבוצה אליה שייך בסיבוכיות משוערכת (למדנו בהרצאות). לאחר מכן, ניגש לעץ הנמצא בסופר פיקסל שמצאנו (שהוא ראש הקבוצה) ונבצע הכנסה\עדכון של למפתח המתאים בעץ בסיבוכיות זמן כאשר הוא כמות התיוגים הקיימים בעץ (הכנסה לעץ בגודל מתבצעת בסיבוכיות ). לכן, סיבוכיות הזמן המשוערכת הכוללת היא

כנדרש.

* : נבדוק את תקינות הפרמטרים שהתקבלו, ונחזיר שגיאות מתאימות במידה ואחד מהן לא תקין בסיבוכיות זמן . נחפש את התמונה בטבלת ה-Hash, ובמידה והיא לא נמצאת נחזיר שגיאה. אחרת, ניגש למבנה הנתונים הנמצא בתמונה, ששומר את קבוצות הסופר פיקסלים השייכים לתמונה. נמצא את הקבוצה אליה שייך בסיבוכיות משוערכת (למדנו בהרצאות). לאחר מכן, ניגש לעץ הנמצא בסופר פיקסל שמצאנו (שהוא ראש הקבוצה) ונבצע הוצאה של המפתח (התיוג) המתאים בעץ בסיבוכיות זמן כאשר הוא כמות התיוגים הקיימים בעץ (הוצאה מעץ בגודל מתבצעת בסיבוכיות ). לכן, סיבוכיות הזמן המשוערכת הכוללת היא

כנדרש.

* : נבדוק את תקינות הפרמטרים שהתקבלו, ונחזיר שגיאות מתאימות במידה ואחד מהן לא תקין בסיבוכיות זמן . נחפש את התמונה בטבלת ה-Hash, ובמידה והיא לא נמצאת נחזיר שגיאה. אחרת, ניגש למבנה הנתונים הנמצא בתמונה, ששומר את קבוצות הסופר פיקסלים השייכים לתמונה. נמצא את הקבוצה (הסופר פיקסל) אליה שייך בסיבוכיות משוערכת (למדנו בהרצאות). ניגש לעץ ה-AVL המצא בסופר פיקסל הזה, שהוא גם עץ דרגות, ונשלוף מהשורש את הניקוד הגבוהה ביותר בעץ בסיבוכיות זמן . לבסוף, נחזיר למשתמש את הערך. לכן סיבוכיות הזמן הכוללת היא .
* : נבדוק את תקינות הפרמטרים שהתקבלו, ונחזיר שגיאות מתאימות במידה ואחד מהן לא תקין בסיבוכיות זמן . נחפש את התמונה בטבלת ה-Hash, ובמידה והיא לא נמצאת נחזיר שגיאה. אחרת, ניגש למבנה הנתונים הנמצא בתמונה, ששומר את קבוצות הסופר פיקסלים השייכים לתמונה. נמצא את 2 הקבוצות (הסופר פיקסלים) אליהן שייכים בסיבוכיות משוערכת  
  (למדנו בהרצאות). אם שני הפיקסלים שייכים לאותו הסופר פיקסל, נחזיר שגיאה. אחרת, נשלוף מ-2 הסופר פיקסלים שמצאנו את עצי הדרגות, ונבצע להם איחוד של עצים בינאריים כפי שלמדנו בתרגולים בסיבוכיות זמן כאשר הוא סכום גדלי העצים. לכן סיבוכיות הזמן המשוערכת היא