

# זבני נתונים 234218 חורף תשע"ט

גיליון רטוב מספר 2 – מעודכן לתאריך 24.12.2018



עמוד 1 מתוך 7

מתרגל ממונה על התרגיל: רועי בר צור, [roi.bar-zur@cs.technion.ac.il](mailto:roi.bar-zur@cs.technion.ac.il)

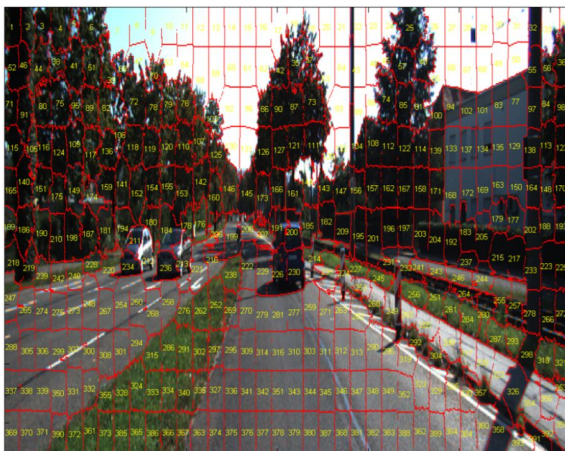
תאריך ושעת הגשה: 12.01.2019 בשעה 23:59

אופן ההגשה: בזוגות. יורד ציון לתרגילים שיוגשו ביחידים בלי אישור מהמתרגל הממונה על התרגיל.

## הנחיות:

- שאלות לגבי דרישות התרגיל יש להפנות באימייל לכתובת הנ"ל.
- תשובות לשאלות המרכזיות אשר ישאלו יתפרסמו בחוץ ה-FAQ באתר הקורס לטובת כלל הסטודנטים.
- שימו לב כי **תוכן ה FAQ הוא מחייב וחובה לקרוא אותו**, אם וכאשר הוא יתפרסם.
- לא** יתקבלו דחיות או ערעורים עקב אי קריאת ה FAQ.
- לפני שאתם ניגשים לקודד את פתרוןכם, ודאו כי יש לכם פתרון העומד **בכל** דרישות הסיבוכיות התרגיל. תרגיל שאינו עומד בדרישות הסיבוכיות יחשב כפסול.
- בתרגיל זה אין הגבלה על מבני הנתונים בהם אתם יכולים להשתמש. מותר וגם מומלץ להשתמש במבנים שמישתם בתרגילים הקודמים, אם הם מתאימים לדרישות הסיבוכיות הנוכחיות.
- העתקת תרגילי בית רטובים תיבדק באמצעות תוכנת בדיקות אוטומטית, המזהה דמיון בין כל העבודות הקיימות במערכת, גם כאלו משנים קודמות. לא ניתן לערער על החלטת התוכנה. התוכנה אינה מבדילה בין מקור להעתק! אנא הימנעו מהסתכלות בקוד שאינו שלכם.**

## הקדמה:



לאחר ההצלחה המסחררת של המערכת החדשה של חברת StaticEye, החברה רוצה את עזרתכם בבניית גרסה חדשה ומשופרת של המערכת.

כעת, אתם נדרשים לבנות מערכת אשר תשמור את המזהים של כל התמונות ועבור כל תמונה תאפשר לאחד קבוצות של פיקסלים לסופר-פיקסלים. המספרים המזהים של התמונות חיוביים.

סופר-פיקסל הוא בעצם איזור בתמונה המכיל אוסף של פיקסלים. כל שני סופר-פיקסלים זרים זה לזה, כלומר, כל פיקסל יכול להשתייך לכל היותר לסופר-פיקסל אחד.

בנוסף, ניתן לתייג כל סופר-פיקסל בתמונה ע"י תיוגים שונים, כאשר לכל תיוג של סופר פיקסל יש ניקוד. כל תיוג מיוצג ע"י מספר מזהה חיובי ייחודי לו.

# זבני נתונים 234218 חורף תשע"ט

גיליון רטוב מספר 2 – מעודכן לתאריך 24.12.2018



עמוד 2 מתוך 7

## הפעולות שבהן מבנה הנתונים צריך לתמוך:

`void * Init(int pixels)`

מאתחל מבנה נתונים ריק.

פרמטרים: pixels מספר הפיקסלים בכל תמונה.

ערך החזרה: מצביע למבנה נתונים ריק או NULL במקרה של כישלון.

סיבוכיות זמן:  $O(1)$  במקרה הגרוע.

**הסיבוכיות של 2 הפעולות הבאות: AddImage, DeletImage היא משוערכת ביחד.**

`StatusType AddImage(void *DS, int imageID)`

הוספת תמונה חדשה עם המזהה `imageID`. כל פיקסל נמצא בסופר-פיקסל נפרד משל עצמו. כל פיקסל ללא תיוגים.

פרמטרים: DS מצביע למבנה הנתונים.

imageID מזהה התמונה שצריך להוסיף.

ערך החזרה: ALLOCATION\_ERROR במקרה של בעיה בהקצאת זכרון.

INVALID\_INPUT אם `DS==NULL` או `imageID<=0`

FAILURE אם `imageID` קיים.

SUCCESS במקרה של הצלחה.

סיבוכיות זמן:  $O(k)$  בממוצע על הקלט משוערך, כאשר  $k$  הוא מספר הפיקסלים בתמונה (pixels).

`StatusType DeletImage(void *DS, int imageID)`

מחיקת התמונה בעלת המזהה `imageID`, הסופר-פיקסלים שלה וכל התיוגים שלהם.

פרמטרים: DS מצביע למבנה הנתונים.

imageID מזהה התמונה שצריך למחוק.

ערך החזרה: ALLOCATION\_ERROR במקרה של בעיה בהקצאת זכרון.

INVALID\_INPUT אם `DS==NULL` או `imageID<=0`

FAILURE אם `imageID` לא קיים.

SUCCESS במקרה של הצלחה.

סיבוכיות זמן:  $O(k + m)$  בממוצע על הקלט משוערך, כאשר  $k$  הוא מספר הפיקסלים בתמונה (pixels) ו- $m$

הוא מספר התיוגים בתמונה בסה"כ.



הסיבוכיות של 4 הפעולות הבאות: **SetLabelScore**, **ResetLabelScore**, **GetHighestScoreLabel**, **MergeSuperPixels** היא משוערכת ביחד.

**StatusType** **SetLabelScore**(**void** \*DS, **int** imageID, **int** pixel, **int** label, **int** score)

תיוג הסופר-פיקסל שמכיל את הפיקסל בעל המזהה pixel בתמונה imageID בתיוג label עם הניקוד score. אם התיוג כבר קיים אז יש לעדכן את הניקוד.

**פרמטרים:** DS מצביע למבנה הנתונים.  
imageID מזהה התמונה.  
pixel מזהה הפיקסל ששייך לסופר פיקסל שיתיוג.  
label התיוג של הפיקסל.  
score הניקוד של התיוג.  
**ערך החזרה:** ALLOCATION\_ERROR במקרה של בעיה בהקצאת זכרון.  
INVALID\_INPUT אם DS==NULL, pixel<0, pixel>=pixels, imageID<=0 או label<=0 או score<=0.  
FAILURE אם לא קיימת תמונה עם מזהה imageID.  
SUCCESS במקרה של הצלחה.  
**סיבוכיות זמן:**  $O(\log^* k + \log m)$  בממוצע על הקלט משוערך, כאשר k הוא מספר הפיקסלים בתמונה (pixels) ו-m הוא מספר התיוגים בסופר-פיקסל המכיל את הפיקסל.

**StatusType** **ResetLabelScore**(**void** \*DS, **int** imageID, **int** pixel, **int** label)

**מחיקת** התיוג label של בסופר-פיקסל המכיל את הפיקסל בעל המזהה pixel בתמונה imageID. לאחר פעולה זו התיוג לא קיים יותר בתמונה בסופר-פיקסל בכלל.

**פרמטרים:** DS מצביע למבנה הנתונים.  
imageID מזהה התמונה.  
pixel מזהה הפיקסל ששייך לסופר פיקסל שיאופס התיוג שלו.  
label התיוג שיש לאפס.  
**ערך החזרה:** ALLOCATION\_ERROR במקרה של בעיה בהקצאת זכרון.  
INVALID\_INPUT אם DS==NULL, pixel<0, pixel>=pixels, imageID<=0 או label<=0.  
FAILURE אם לא קיימת תמונה עם מזהה imageID או שהסופר-פיקסל של pixel לא מתיוג ע"י label.  
SUCCESS במקרה של הצלחה.  
**סיבוכיות זמן:**  $O(\log^* k + \log m)$  בממוצע על הקלט משוערך, כאשר k הוא מספר הפיקסלים בתמונה (pixels) ו-m הוא מספר התיוגים בסופר-פיקסל המכיל את הפיקסל.

# זבני נתונים 234218 חורף תשע"ט

גיליון רטוב מספר 2 – מעודכן לתאריך 24.12.2018



עמוד 4 מתוך 7

**StatusType** GetHighestScoreLabel(void \*DS, int imageID, int pixel, int \*label)

מציאת התיג בעל הניקוד הכי גבוה בסופר-פיקסל המכיל את pixel בתמונה imageID. במקרה של שוויון יש להחזיר את התיג בעל המזהה הגדול יותר.

פרמטרים:	DS	מצביע למבנה הנתונים.
	imageID	מזהה התמונה.
	pixel	מזהה הפיקסל ששייך לסופר-פיקסל הרלוונטי.
	label	מצביע למשתנה שיעודכן למזהה של התיג עם הניקוד הכי גבוה.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם DS==NULL, pixel<0, pixel>=pixels, imageID<=0 או label==NULL.
	FAILURE	אם לא קיימת תמונה עם מזהה imageID או שהסופר-פיקסל של pixel לא מתויג בכלל.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log^* k)$	במוצק על הקלט משוערך, כאשר k הוא מספר הפיקסלים בתמונה (pixels).

**StatusType** MergeSuperPixels(void \*DS, int imageID, int pixel1, int pixel2)

איחוד הסופר-פיקסלים של הפיקסלים pixel1 ו-pixel2 לסופר-פיקסל אחד בתמונה עם המזהה imageID. הסופר-פיקסל החדש מתויג על ידי כל התיגים של הסופר-פיקסלים המקוריים לפי סכום הניקודים של התיג בסופר-פיקסלים המקוריים. כלומר, אם הסופר-פיקסל הראשון היה מתויג ע"י התיג "20" עם ניקוד 5 והסופר-פיקסל השני היה מתויג ע"י התיג "20" עם ניקוד 3 אז הסופר-פיקסל החדש צריך להיות מתויג ע"י התיג "20" עם ניקוד 8.

פרמטרים:	DS	מצביע למבנה הנתונים.
	imageID	מזהה התמונה.
	pixel1	מזהה הפיקסל הראשון.
	pixel2	מזהה הפיקסל השני.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם DS==NULL, imageID<=0, pixel1<0, pixel1>=pixels, pixel2<0 או pixel2>=pixels.
	FAILURE	אם imageID לא קיים או ששני הפיקסלים שייכים לאותו הסופר-פיקסל.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log^* k + m)$	במוצק על הקלט משוערך, כאשר k הוא מספר הפיקסלים בתמונה ו-m (pixels) הוא מספר התיגים בשני הסופר-פיקסלים הרלוונטיים.

# זבני נתונים 234218 חורף תשע"ט

גיליון רטוב מספר 2 – מעודכן לתאריך 24.12.2018



עמוד 5 מתוך 7

`void Quit(void **DS)`

הפעולה משחררת את המבנה. בסוף השחרור יש להציב ערך NULL ב-DS, אף פעולה לא תקרא לאחר מכן

פרמטרים: DS מצביע למבנה הנתונים.

ערך החזרה: אין.

סיבוכיות זמן:  $O(n \cdot k \cdot m)$  במקרה הגרוע, כאשר n הוא מספר התמונות, k הוא מספר הפיקסלים בכל

תמונה ו-m הוא מספר התיגים השונים הקיימים במערכת.

סיבוכיות מקום:  $O(n \cdot k \cdot m)$  במקרה הגרוע, כאשר n הוא מספר התמונות, k הוא מספר הפיקסלים בכל תמונה

ו-m הוא מספר התיגים השונים הקיימים במערכת.

## ערכי החזרה של הפונקציות:

בכל אחת מהפונקציות, ערך ההחזרה שיוחזר ייקבע לפי הכלל הבא:

- תחילה, יוחזר INVALID\_INPUT אם הקלט אינו תקין.
- אם לא הוחזר INVALID\_INPUT:
  - בכל שלב בפונקציה, אם קרתה שגיאת הקצאה יש להחזיר ALLOCATION\_ERROR.
  - אם קרתה שגיאה אחרת, כפי שמצוין בכל פונקציה, יש להחזיר מיד FAILURE מבלי לשנות את מבנה הנתונים.
- אחרת יוחזר SUCCESS.

## חלק יבש:

- **הציון על החלק היבש הוא 50% מציון התרגיל.**
- לפני מימוש הפעולות בקוד יש לתכנן היטב את מבני הנתונים והאלגוריתמים ולוודא כי באפשרותכם לממש את הפעולות בדרישות הזמן והזיכרון שלעיל.
- מומלץ לממש תחילה את מבני הנתונים בצורה הכללית ביותר ורק אז לממש את הפונקציות הנדרשות.
- יש להכין מסמך הכולל תיאור של מבני הנתונים והאלגוריתמים בהם השתמשתם בצירוף הוכחת סיבוכיות הזמן והמקום שלהם. חלק זה עומד בפני עצמו וצריך להיות מובן לקורא גם לפני העיון בקוד. אין צורך לתאר את הקוד ברמת המשתנים, הפונקציות והמחלקות, אלא ברמה העקרונית.
- ראשית הציגו את מבני הנתונים בהם השתמשתם. רצוי ומומלץ להיעזר בציור.
- לאחר מכן הסבירו כיצד מימשתם כל אחת מהפעולות הנדרשות. הוכיחו את דרישות סיבוכיות הזמן של כל פעולה תוך כדי התייחסות לשינויים שהפעולות גורמות במבני הנתונים.
- הוכיחו שמבנה הנתונים וכל הפעולות עומדים בדרישת סיבוכיות המקום.
- רמת פירוט: יש להסביר את כל הפרטים שאינם טריוויאליים ושחשובים לצורך מימוש הפעולות ועמידה בדרישות הסיבוכיות. אין לדון בפרטים טריוויאליים (הפעילו את שיקול דעתכם בקשר לזה, ושאלו את האחראי על התרגיל אם אינכם בטוחים). אין לצטט קטעים מהקוד כתחליף להסבר. אין צורך לפרט

# זבני נתונים 234218 חורף תשע"ט

גיליון רטוב מספר 2 – מעודכן לתאריך 24.12.2018



## עמוד 6 מתוך 7

אלגוריתמים שנלמדו בכתה. כמו כן, אין צורך להוכיח תוצאות ידועות שנלמדו בכתה, אלא מספיק לציין בבירור לאיזו תוצאה אתם מתכוונים.

- הגשת החלק הרטוב מהווה תנאי הכרחי לקבלת ציון על החלק היבש, כלומר, הגשה בה יתקבל אך ורק חלק יבש תגרור ציון 0 על התרגיל כולו.
- על חלק זה לא לחרוג מ-8 עמודים.

### חלק רטוב:

- אנו ממליצים בחום על מימוש **Object Oriented**, ב++C. על מנת לעשות זאת הגדירו מחלקה, נאמר **ImageTagger**, וממשו בה את דרישות התרגיל. אח"כ, על מנת לייצר התאמה לממשק ה C ב **library.h**, ממשו את **library.cpp** באופן הבא:

```
#include "library.h"
#include "ImageTagger.h"

void* init(int pixels) {
    ImageTagger* DS = new ImageTagger(pixels);
    return (void*)DS;
}

StatusType AddImage(void* DS, int imageID) {
    return ((ImageTagger*)DS) -> AddImage(imageID);
}
```

- על הקוד להתקמפל על 2 באופן הבא:

**g++ -std=c++11 -DNDEBUG -Wall \*.cpp**

עליכם מוטלת האחריות לוודא קומפילציה של התכנית ב ++g. אם בחרתם לעבוד בקומפיילר אחר, מומלץ לקמפל ב g++ מידי פעם במהלך העבודה.

### הערות נוספות:

- חתימות הפונקציות שעליכם לממש ומספר הגדרות נמצאים בקובץ **library.h**.
- קראו היטב את הקובץ הנ"ל, לפני תחילת העבודה.
- אין לשנות את הקבצים הנ"ל ואין להגיש אותם.
- עליכם לממש בעצמכם את כל מבני הנתונים (אין להשתמש במבנים של STL ואין להוריד מבני נתונים מהאינטרנט).
- יש לתעד את הקוד בצורה נאותה וסבירה.
- מסופקת לכם דוגמא של קובץ קלט (**in.txt**) וקובץ הפלט (**out.txt**) המתאים לו.

# זבני נתונים 234218 חורף תשע"ט

גיליון רטוב מספר 2 – מעודכן לתאריך 24.12.2018



עמוד 7 מתוך 7

- שימו לב: התוכנית שלכם תיבדק על קלטים שונים מקבצי הדוגמא הנ"ל, שיהיו ארוכים ויכללו מקרי קצה שונים. לכן, מומלץ מאוד לייצר בעצמכם קבצי קלט ארוכים, לבדוק את התוכנית עליהם, ולוודא שהיא מטפלת נכון בכל מקרה הקצה.

## הגשה:

### חלק יבש+ חלק רטוב:

- הגשת התרגיל הנה אך ורק אלקטרונית דרך אתר הקורס.
- יש להגיש קובץ ZIP (ללא תיקיות או תתי תיקיות בתוכו) שמכיל את הדברים הבאים:
  - קבצי ה-Source Files שלכם (ללא הקבצים שפורסמו).
  - קובץ PDF אשר מכיל את הפתרון היבש. מומלץ להקליד את החלק הזה. ניתן להגיש קובץ PDF מבוסס על סריקה של פתרון כתוב בכתב יד. שימו לב כי במקרה של כתב לא קריא, כל התרגיל לא ייבדק.
  - קובץ submissions.txt, המכיל בשורה הראשונה את שם, תעודת הזהות וכתובת הדוא"ל של השותף הראשון ובשורה השנייה את שם, תעודת הזהות וכתובת הדוא"ל של השותף השני. לדוגמה:

Jane Roe 012345678 [janeroe@cs.technion.ac.il](mailto:janeroe@cs.technion.ac.il)

John Doe 123456789 [johndoe@cs.technion.ac.il](mailto: johndoe@cs.technion.ac.il)

- שימו לב כי אתם מגישים את כל שלושת החלקים הנ"ל.
- אין להשתמש בפורמט כיווץ אחר, מאחר ומערך הבדיקה האוטומטי אינו יודע לזהות פורמטים אחרים.
- אין להגיש קובץ המכיל תתי תיקיות.
- לאחר שהגשתם, יש באפשרותכם לשנות את התוכנית ולהגיש שוב.
- ההגשה האחרונה היא הנחשבת.
- הגשה שלא תעמוד בקריטריונים הבאים תפסל ותיקנס בנקודות!

## דחיות ואיחורים בהגשה:

- דחיות בתרגיל הבית תינתנה אך ורק לפי תקנון הקורס.
- 5 נקודות יורדו על כל יום איחור בהגשה ללא אישור מראש. באפשרותכם להגיש תרגיל באיחור של עד 5 ימים ללא אישור. תרגיל שיוגש באיחור של יותר מ-5 ימים ללא אישור מראש יקבל 0.
- במקרה של איחור בהגשת התרגיל יש עדיין להגיש את התרגיל אלקטרונית דרך אתר הקורס.
- במקרה של איחור מוצדק, יש לצרף לקובץ ה PDF שלכם את סיבות ההגשה באיחור, לפי הטופס המופיע באתר, כולל סריקות של אישורי מילואים או אישורי נבחן.

**בהצלחה!**