



## תרגיל בית 5 – מבנה נתונים שווה שטח ושווה נתונים

### נתונים

1. מסד נתונים "קטן" (Little) מתרגיל הבית הקודם. זהו מסד נתונים יחסית קטן עמו נח לעבוד ולבצע את החיפוש ו-Debugging.
2. מסד נתונים "גדול" (Big) עליו תתבצע בדיקת המתרגל – והוא נרחב יותר בכמות הרשומות. עליכם לוודא כי התוכנה שלכם מצליחה להציג ולבצע את המשימות בתוכו.

### מטרות התרגיל

בתרגיל זה נרצה למצוא חלקות הנמצאות ברדיוס מסוים מנקודה כלשהי כאשר חלקה נחשבת בתוך רדיוס החיפוש אם לפחות אחת מנקודות המפנה שלה נמצאות בתוך רדיוס החיפוש. החיפוש יתאפשר לפי מעבר על כל רשימת הנקודות וגם על ידי אחת מהשיטות הבאות (חלוקה תתבצע בכיתה):

1. חיפוש תוך שימוש במבנה נתונים שווה שטח.
2. חיפוש תוך שימוש במבנה נתונים שווה נתונים.

את החיפוש בשיטות השונות יש לבצע בזיכרון המחשב, כלומר יש לטעון את הטבלאות המתאימות לזיכרון המחשב ורק לאחר מכן לבנות את מבני הנתונים הנדרשים.

### חלק ראשון - חיפוש חלקות ובעליהן:

עליכם לכתוב תוכנית בפייתון בעלת ממשק גרפי על פי ההנחיות הבאות:

1. עם הפעלת התוכנית, יתבקש המשתמש לבחור את קובץ מסד הנתונים בעזרת דיאלוג מתאים (אין להניח שהקלט תקין ויש לאפשר בחירה של מסד אחר ללא צורך לסגור את התוכנית).
2. לאחר בחירת מסד הנתונים תטען רשימת הנקודות (וכל רשימה נדרשת אחרת) מתוך מסד הנתונים לזיכרון המחשב ותישמר במבנה נתונים מתאים אשר יאפשר לבצע חיפוש בכל אחת מן השיטות שהוזכרו מעלה.
3. הממשק יאפשר הכנסת קואורדינטות הנקודה סביבה יתבצע החיפוש, את רדיוס החיפוש ואת השיטה בה יתבצע החיפוש. כמו כן תתאפשר הזנת הפרמטרים הבאים:
  - א. עבור מבנה נתונים שווה שטח: מספר הנקודות הממוצע לתא (בשם: average value), והיחס בין גובה ורוחב התא (בשם: ratio).
  - ב. עבור מבנה הנתונים שווה נתונים: מספר הנקודות המקסימאלי עבור כל תא (בשם: maximum points).

הערה: לא ניתן להניח שהקלט תקין בשלב זה – יש להציג למשתמש הודעה במקרה של קלט לא תקין.

4. בלחיצה על כפתור (Browse), התוכנה תחפש את כל החלקות אשר נמצאות בתוך רדיוס החיפוש.



5. תוצאות החיפוש תוצגנה בשלושה אופנים :

- א. רשימה של החלקות בעזרת פקד `listbox`. ברשימה, יש להציג את שם החלקה ומספר ושם הגוש אליו שייכת. לחיצה על החלקה ברשימה יכתוב בשדה טקסט נפרד את שטחה, מספר התצ"ר בו נמצאת החלקה ומספר המודד שביצע את התצ"ר.
- ב. רשימה של האינדקסים של התאים עבור חלוקה של מבנה נתונים שווה שטח, וערך הרמה הכי גדולה עבור חיפוש במבנה נתונים שווה נתונים. רשימות אלה יהיו ריקות בשיטת החיפוש העוברת על כל רשימת הנקודות (השיטה הראשונה).
- ג. תצוגה גרפית של תוצאות החיפוש : יש להציג את הנקודה, רדיוס החיפוש וכל החלקות הנמצאות בתוך רדיוס זה כאשר כל אחד מסוגי הפרטים יוצג בצבע שונה (גבולות החלקות יוצגו בצבעים זהים. חלקה שתיבחר ב `listbox` תיציג בצבע שונה). כמו כן, יש לשרטט את כל גבולות התאים (של מבנה נתונים שווה שטח או שווה נתונים – בהתאם לבחירת המבנה) בהם נמצאות נקודות של החלקות בתוך רדיוס החיפוש.

6. כמו כן, יש להציג את זמני הריצה עבור חיפוש החלקות. כאשר מדובר בזמן ריצת החיפוש בלבד ללא הפעלת שאילתות SQL בהן בחרתם להשתמש לצורך מילוי רשימת החלקות.

## הנחיות למימוש

1. עליכם לממש מחלקה לתוכנה שלכם שתכיל את הממשק הגרפי והמתודות הראשיות. בין מתודות אלה החיפוש הפשוט מתוך רשימה של נקודות בעזרת `list`. החיפוש יבוצע בצורה טריוויאלית בעזרת מעבר על כל הרשימה ובדיקה מי מתאים לרדיוס בריבוע (בכדי לחסוך את חישוב השורש).
  - א. את מחלקה זו עליכם לתכנן בעצמכם.
  - ב. המתודה תמומש עם החתימה הבאה : `browsePointsInRadius (thePnt (2D tuple, inRadius (float)`. המתודה תחזיר רשימת נקודות המכילה את כל הנקודות הנמצאות בתוך או על רדיוס `inRadius` מהנקודה `thePnt`.
2. עליכם לכתוב את המחלקה `GeoGridCells` המכילה תאים עם הפניות לנקודות (הכוונה לפרנס לנקודות ברשימת הנקודות – לא להכיל בתא את הנקודות עצמן), מבנה התאים הינו שווה שטח כפי שנלמד בכיתה.
  - א. את המחלקה `GeoGridCells` עליכם לתכנן בעצמכם.
  - ב. עליכם לכתוב קונסטרקטור מתאים לבניית מבנה נתונים זה.
  - ג. עליכם להוסיף למחלקה את המתודה בעלת החתימה הבאה : `browsePointsInRadius (thePnt (2D tuple, inRadius (float)`. המתודה תחזיר רשימת נקודות המכילה את כל הנקודות הנמצאות בתוך או על רדיוס `inRadius` מהנקודה `thePnt`. החיפוש יבוצע ע"י האלגוריתם למציאת נקודות ברדיוס מסוים במבנה נתונים שווה שטח.
3. עליכם לכתוב את המחלקה `GeoQuadtreeCells` המכילה תאים של `QuadTree` עם הפניות לרשימת הנקודות (כמו בסעיף הקודם) מתוך העלים. במקרה שהתא אינו עלה, הרשימה ריקה. מבנה העץ `QuadTree` כפי שלמדנו בכיתה.
  - א. את המחלקה `GeoQuadtreeCells` עליכם לתכנן בעצמכם.
  - ב. עליכם לכתוב קונסטרקטור מתאים לבניית מבנה נתונים זה.



ג. עליכם להוסיף למחלקה את המתודה בעלת החתימה הבאה:

`browsePointsInRadius (thePnt (2D tuple), inRadius (float))`

המתודה תחזיר רשימת נקודות המכילה את כל הנקודות הנמצאות בתוך או על רדיוס `inRadius` מהנקודה `thePnt`. החיפוש יבוצע ע"י האלגוריתם למציאת נקודות ברדיוס מסוים במבנה נתונים `QuadTree`.

4. לאחר שהחלקות נמצאו, מומלץ להריץ שאילתה (או רצף שאילתות) המוצאת את הפוליון של כל חלקה ומכניסה אותם לרשימת פוליונים בכדי להציג אותם על המסך. **אין צורך לממש מחלקת פוליון**. פוליון יכול להיות מיוצג על ידי רשימת נקודות.

**שימו לב כי ניתן להוסיף מתודות או מחלקות אם אתם רואים בכך צורך אך חובה לממש את המחלקות והמתודות המוגדרות כאן ואין לשנות את שמות המחלקות וההגדרות שלהן.**

### חלק שני - בדיקות ביצועים:

בחלק זה עליכם לחקור את מגבלות השיטות אותן ממשתם. בכדי להגיע למסקנה, בצעו את השלבים הבאים:

1. תכננו והריצו אוסף שאילתות חיפוש בנקודות שונות עם רדיוס קבוע. עבור כל הרצה:  
א. במבנה נתונים שווה שטח: שנו את מספר הנקודות הממוצע לתא. יחס גובה / רוחב התא יהיה שווה ליחס גובה / רוחב המלבן החוסם של כלל הנקודות.  
ב. במבנה נתונים שווה נתונים: שנו את מספר הנתונים בתא.  
2. יש לבצע את אותה הרצה 3 פעמים ולעבוד עם הזמן הממוצע.  
הערה: את השלב הזה מומלץ לעשות כאשר שום דבר אחר מלבד התוכנית שלכם לא רץ ברקע, את ההרצות כדאי לעשות בהרצה ולא ב- debug בכדי למנוע הפרעות מיותרות בזמן ההרצה, את כל ההרצות מומלץ לעשות באותו זמן ולא בימים שונים בכדי לסנן גורמים מפריעים.
3. הכינו טבלה המסכמת את תוצאות ההרצה עבור נקודות שונות, רדיוס חיפוש קבוע והפרמטר באמצעותו הגדרתם את מבנה הנתונים. טבלה לדוגמה, עבור מבנה נתונים שווה שטח:



תוצאות חיפוש עבור מבנה נתונים שווה שטח וגודל רדיוס חיפוש $r$					
נקודה	מס' נקודות ממוצע לתא	מס' הרצה	חיפוש שווה שטח	חיפוש פשוט	שווה נתונים
$(x_1, y_1)$	$n_1$	1			
		2			
		3			
		ממוצע			
	$n_2$	1			
		2			
		3			
		ממוצע			
	⋮	⋮			
$(x_2, y_2)$	$n_1$				
	$n_2$				
	⋮				

4. הכינו גרף המתאר את זמן החיפוש כפונקציה של מספר הנקודות לתא עבור כל אחת מהשיטות. עבור כל מספר יש לקחת את כלל הזמנים שחושבו איתו ולחשב זמן חיפוש ממוצע.
5. הגיעו למסקנות לגבי מגבלות מבנה הנתונים. עבור איזה ערך לפרמטר מבנה זה יעיל ביותר עבור מסד הנתונים איתו עבדתם?



## הנחיות להגשה

1. יש להגיש את התרגיל עד ה- 20.6.19 בשעה 23:59.
2. הגשת התרגיל הינה ביחידים
3. עליכם להגיש שני קבצים:
  - א. קובץ בפורמט pdf המכיל:
  - i. את התיעוד החיצוני. על התיעוד החיצוני להכיל הנחיות הפעלה לתוכנית, דיאגרמה של המחלקות שמומשו, הסבר לאלגוריתמים בהם השתמשתם. אין צורך לכתוב אלגוריתמים שהוסברו בכיתה, ניתן לכתוב השתמשנו באלגוריתם ABC כפי שנלמד, יש לציין איזו שיטה בחרתם ושינויים שעשיתם או אלגוריתמים שאתם כתבתם. **על התיעוד להיות לא יותר משני עמודים בגודל פונט קריא (כלומר גודל פונט של 12 ומעלה) – לא כולל הדיאגרמה.**
  - ii. דוח בדיקת הביצועים המכיל את התכנון והסבר מלא כיצד בחרתם את הנקודות והפרמטרים, סיכום התוצאות והמסקנה אליה הגעתם וכיצד הגעתם אליה. היקף הדוח הוא לכל היותר שלושה עמודים בגודל פונט קריא כולל האיורים (הגרפים) המצורפים. כמובן שעל הגרפים להיות ברורים וקריאים.
  - iii. תחת הכותרת 'נספחים': הקוד המלא של המחלקות הבאות אותן מימשתם: GeoGridCells, GeoQuadtreeCells, המחלקה הראשית של התוכנה וכל מחלקה אחרת אותה בחרתם לממש. יש לסדר את הקוד תחת תתי כותרות מתאימות.
- ב. ספריית הפרויקט עם כל קבצי הקוד מאורגנים בתיקיות המתאימות. אין לצרף להגשה את קובץ מסד הנתונים. עליכם לתעד את הקוד שרשמתם (תיעוד פנימי) בצורה ברורה ותמציתית באנגלית בלבד, לפי ההנחיות הבאות:
  - i. כל מחלקה (class) תופיע בקובץ נפרד.
  - ii. בתחילת כל מחלקה או מתודה יהיה הסבר קצר על מטרות המחלקה/מתודה והסבר על הקלט והפלט של המתודות ושיטת המימוש (האלגוריתם שבו השתמשתם במתודה). הנחייה זו תקפה גם עבור המחלקות והמתודות שנדרש מכם לממש בצורה מפורשת ועבור המתודות המטפלות באירועים של הממשק.
  - iii. כל שדה ילווה בשורת הסבר.
  - iv. עבור כל לולאה יש לרשום בקצרה את מטרתה, על אילו אובייקטים הלולאה עוברת ומה מתרחש בתוכה.
  - v. עבור כל קטע קוד משמעותי יש לרשום הסבר קצר.
4. את שני הקבצים יש לקבץ לקובץ rar/zip אחד בשם ות"ז של הסטודנט
5. שימו לב כי **תוכנית שלא תרוץ לא תיבדק**. לפני ההגשה, מומלץ לבדוק שהתוכנית עובדת על מחשב אחר.
6. לתרגיל ייתכן שתתבצע בדיקה פרונטאלית בתאריך ומיקום שייקבע בהמשך. עליכם להגיע לבדיקה במהלך השעות שיקבעו. הבדיקה **פרונטאלית הינה חובה** והציון ייקבע לאחריה.

## בהצלחה