

## מבני נתונים – פרויקט מספר 2 – גרף עם משקלי

### סביבות

#### תיאור הבעיה

עליכם לממש מבנה נתונים המייצג גרף פשוט לא מכוון (קודקודים + קשתות). לכל קודקוד בגרף זה יש מזהה ומשקל (שניהם מספרים שלמים ואי-שליליים), ועליכם יהיה לתחזק מבנה נתונים שמאפשר לבצע שינויים בגרף כנדרש, וגם להחזיר באופן יעיל את הקודקוד שסכום משקלו ומשקלי שכניו הוא המקסימלי כרגע (בדומה לשאלה שקיבלתם בתרגיל 3 בשיעורי הבית).

#### נוטציה

הנוטציה בתרגיל זה היא:

- $N$  הוא מספר הצמתים במצב ההתחלתי של הגרף (לאחר ה-constructor)
- $n$  הוא מספר הצמתים בגרף כרגע.
- $m$  הוא מספר הקשתות בגרף כרגע.

בנוסף, עבור צומת  $v$  נסמן ב- $w_v$  את משקל הצומת  $v$  וב- $\mu(v)$  את שכניו של  $v$ . נגדיר את משקל הסביבה של  $v$  להיות  $w_v + \sum_{v' \in \mu(v)} w_{v'}$  ונסמנו ב- $B_v$ .

#### תיאור המבנה והפעולות

על מבנה הנתונים לצרוך לא יותר מ- $O(m + N)$  זיכרון בכל רגע נתון.

הפעולות שיש לממש הן (הסיבוכיות הנדרשת של כל הפעולות היא בתוחלת):

1. `Graph(Node [] nodes)` – מקבלת מערך צמתים מאורך  $N$  (כאשר כל אובייקט צומת מכיל את המזהה שלו ואת המשקל שלו) ומאתחלת את מבנה הנתונים כגרף עם  $N$  צמתים אלו וללא קשתות.  
ניתן להניח שלשני קודקודים שונים יהיה מזהה שונה.  
סיבוכיות זמן נדרשת:  $O(N)$ .
2. `addEdge(int node1_id, int node2_id)` – מקבלת שני מזהים של צמתים. אם שני הצמתים נמצאים בגרף, הפעולה מוסיפה קשת ביניהם. אחרת, היא איננה עושה דבר. ניתן להניח שלעולם לא תתבקשו להוסיף קשת בין שני צמתים בגרף שכבר קיימת ביניהם קשת.  
הפונקציה מחזירה `true` אם נוספה קשת ואחרת מחזירה `false`.  
סיבוכיות זמן נדרשת:  $O(\log n)$ .
3. `maxNeighborhoodWeight()` – מחזירה את הצומת  $v$  כך שמשקל הסביבה שלו  $B_v$  הוא מקסימלי. אם בגרף אין צמתים, מחזירה `null`.  
סיבוכיות זמן נדרשת:  $O(1)$ .
4. `getNeighborhoodWeight(int node_id)` – בהינתן מזהה קודקוד `node_id`, מחזיר את משקל הסביבה של הקודקוד בעל מזהה זה אם הוא בגרף. אם אין קודקוד בעל מזהה זה בגרף, יש להחזיר `-1`.  
סיבוכיות זמן נדרשת:  $O(1)$ .

5. `deleteNode(int node_id)` – מוחקת את הקודקוד  $v$  בעל המזהה `node_id` (ואת כל הקשתות המערבות אותו) מהגרף, אם הוא קיים (אחרת, הפעולה לא מבצעת דבר). שימו לב שמשקלו של קודקוד שנמחק אינו נספר יותר במשקל הסביבה של אף קודקוד אחר. הפונקציה מחזירה `true` אם נמחק צומת, ואחרת `false`.  
סיבוכיות זמן נדרשת:  $O((d_v + 1) \log n)$ , כאשר  $d_v$  היא דרגתו של הצומת  $v$  (כלומר, מספר שכניו בגרף כרגע).
6. `getNumNodes()` מחזירה את מספר הצמתים שנמצאים בגרף כרגע.
7. `getNumEdges()` מחזירה את מספר הקשתות שנמצאות בגרף כרגע.

## הנחות עבודה

- ניתן להניח שכל המזהים של קודקודים הם מספרים שלמים ואי-שליליים שהם לכל היותר  $10^9$ , ושהמזהים של שני קודקודים שונים הם שונים.
- המשקלים של צמתים הם תמיד מספרים שלמים ואי-שליליים.

## פרטי מימוש

הפתרון שלכם חייב להכיל את הרכיבים הבאים:

- טבלת `hash` הממפה ממזהה קודקוד לייצוגו בגרף ובערימת המקסימום. עליכם לממש את טבלת `hash` עם `universal hashing` ו-`chaining` עם משפחת פונקציות המודולו, כאשר הראשוני הרלוונטי הוא  $p = 10^9 + 9$ .
  - מבנה של גרף בפורמט "רשימת שכנויות" כאשר רשימת השכנים של כל צומת תהיה רשימה מקושרת דו-כיוונית, ובנוסף יתוחזקו מצביעים דו-כיווניים בין שני ייצוגיה של כל קשת  $(u, v)$  (הייצוג שלה ב- $u$  והייצוג שלה ב- $v$ ).
  - ערימת מקסימום בה כל זוג מפתח-ערך הוא כך שהמפתח הוא סכום הסביבה של צומת כלשהו  $u$ , והערך הוא הצומת  $u$ .
- הפרטים במימוש המבנה הם לבחירתכם (לדוגמה, האם לשמור כערך בערימה מזהה של קודקוד או פוינטר של הקודקוד עצמו), אך יש להשתמש במבני נתונים אלו לפתרון השאלה. כמובן שעליכם לממש את מבני הנתונים האלה בעצמכם, ואין להשתמש בספריה חיצונית.

## סיבוכיות

יש לתעד בקוד ובמסמך נפרד (ביותר פירוט) את סיבוכיות זמן הריצה בתוחלת (האסימפטוטית, במונחי  $O$  הדוקים) של כל פונקציה, כתלות במספר האיברים בעץ  $n$ . עליכם להשיג סיבוכיות זמן ריצה (במקרה הגרוע ביותר) נמוכה ככל הניתן עבור כל אחת מהפונקציות, והיכן שמצויינת דרישת סיבוכיות עליכם כמובן לעמוד בה. במסמך הנפרד עליכם להצדיק סיבוכיות של פעולות אם הניתוח איננו טריוויאלי.

## פלט

אין צורך בפלט למשתמש.

## תיעוד

### תיעוד קוד:

בנוסף לבדיקות אוטומטיות של הקוד שלכם, קובץ המקור ייבדק גם באופן ידני. חשוב להקפיד על תיעוד לכל פונקציה, וכמות סבירה של הערות. הקוד צריך להיות קריא, בפרט הקפידו על בחירת שמות משתנים, על אורך השורות, מבנה פשוט לקוד וכו'.

### תיעוד חיצוני:

ככתוב מעלה, יש להגיש בנוסף לקוד גם מסמך תיעוד חיצוני. המסמך יכול את:

- תיאור כל המחלקות שמימשתם ותפקידיהן.
- עבור כל מחלקה, תיאוריהם של כל חברי/מתודות המחלקה.
- עבור מתודות, תיעוד וניתוח סיבוכיות (כמפורט מעלה), והתייחסות למטרתה ולדרך פעולתה. שימו לב שדרישת התיעוד הזו תקפה גם לגבי פונקציות עזר.

## בדיקות

התרגילים ייבדקו באמצעות תוכנת טסטר שקוראת לפונקציות המפורטות מעלה, ומוודאת את נכונות התוצאות. קובץ הטסטר שלנו **לא יפורסם** לפני הבדיקות. עליכם לבדוק את המימוש בעצמכם! בפרט, כדאי מאוד לממש טסטר, כדי לבדוק את תקינות ונכונות המימוש, כולל במקרי קצה שונים.

**חשוב:** בקובץ שתגישו לא תהיה פונקציית main (דבר זה יפגע בטסטר שיבדוק לכם את התרגילים). אם הצלחתם לקמפל את הפרויקט לבדו (ללא טסטר), זה סימן שמשהו לא נכון במימוש שלכם.

הקוד ייבדק על מחשבי בית הספר על גירסא Java8.

הנחיות להשמשת סביבת העבודה בבית (ג'אוה+אקליפס):

<http://courses.cs.tau.ac.il/software1/1415b/misc/workenv.pdf>

מדריך לעבודה עם Eclipse (סעיפים 5-9, 15):

<http://www.vogella.com/>

הנחיות לפתיחת חשבון מחשב, למי שמעוניין/ת לעבוד במעבדת בית הספר:

<http://cs.tau.ac.il/system/accounts0>

שימוש בג'אוה 8 במעבדות האוניברסיטה:

<http://courses.cs.tau.ac.il/software1/1415b/misc/lab-eclipse.pdf>

## מידות

בתרגיל זה ננצל את המבנה שמימשתם על מנת למדוד את הדרגה המקסימלית בגרף אקראי. ישנן כמה וכמה מודלים לגרף כזה, ואנחנו נתייחס לאחד הבסיסיים והמפורסמים שנקרא גרף Erdos-Renyi, בו כל קשת מופיעה בהסתברות שווה. עוד על מודל זה לגרפים אקראיים ניתן לקרוא כאן: [https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi\\_model](https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi_model)

עבור כל  $i$  טבעי בין 6 ל-21 (כולל), בצעו את הניסוי הבא:

- צרו גרף בן  $n = 2^i$  צמתים.
- הוסיפו לגרף  $n$  קשתות אקראיות מתוך  $\binom{n}{2}$  הקשתות האפשריות, כך שכל הקשתות סבירות באותה המידה (איך תוכלו לממש זאת בזמן סביר מבלי להסתכן בלהכניס את אותה הקשת פעמיים?)
- השתמשו ב-`maxNeighborhoodWeight` וב-`getNeighborhoodWeight` על מנת למצוא את הדרגה המקסימלית של קודקוד בגרף (מה הקשר בינה לבין משקל הסביבה שלו?). רשמו את הדרגה המקסימלית בטבלה.

מספר סידורי $i$	$n$	דרגה מקסימלית בגרף
6	64	
7	128	
...		
21	2097152	

איך הדרגה המקסימלית גדלה ביחס ל- $n$ ? התבוננו בלינק הבא:

[https://en.wikipedia.org/wiki/Balls\\_into\\_bins\\_problem](https://en.wikipedia.org/wiki/Balls_into_bins_problem)

איך הבעיה שלפניכם קשורה לבעיה שבלינק? האם התוצאות האמפיריות שראיתם מתיישבות עם מה שהייתם מצפים? הסבירו ונמקו.

### הגשה

הגשת התרגיל תתבצע באופן אלקטרוני באתר הקורס במודל.  
**הגשת התרגיל היא בזוגות בלבד!** על המעוניינים לחרוג מכך (להגיש ביחיד, או במקרים מוצדקים בשלישיה) לפנות באופן אישי למתרגלים.  
כל זוג ייבחר נציג **אחד** ויעלה תחת שם המשתמש שלו את קבצי התרגיל (תחת קובץ zip) למודל. על ההגשה לכלול שלושה קבצים:  
קובץ המקור (הרחבה של קובץ השלד שניתן) תחת השם Graph.java.  
קובץ טקסט info.txt המכיל את פרטי המגישים הבאים: תז, שמות ושמות משתמש.  
מסמך תיעוד חיצוני, המכיל גם את תוצאות המדידות. את המסמך יש להגיש באחד הפורמטים הבאים: pdf, doc, docx.  
שמות קובץ התיעוד וקובץ zip צריכים לכלול את שמות המשתמש האוניברסיטאיים של **שני המגישים** לפי הפורמט Graph\_username1\_username2.pdf/doc/zip/... בתוכן הקבצים יש לציין את שמות המשתמש, תעודות הזהות ושמות המגישים (בכותרת המסמך ובשורת הערה בקובץ המקור).

הגשת שיעורי הבית באיחור - באישור מראש בלבד. הגשה באיחור ללא אישור תגרור הורדת נקודות מהציון.  
הגשת התרגיל היא חובה לשם קבלת ציון בקורס.

### בהצלחה!