תכנות ותכן מונחה עצמים – תרגיל 2

מגישים:

גילי קרני 305200420

גל שחף 201285566

# שאלה 1

## סעיף א' – תיעוד חיצוני

בחרנו בפעולות הבאות ע"מ לממש את ההפשטה:

1. אתחל גרף נקי.
2. הוספת צומת עם label (שחור/לבן)
3. הוספת קשת עם label
4. החזר רשימת צמתים (שחורים/לבנים)
5. החזר רשימת labels של הורים של צומת לפי label
6. החזר רשימת labels של ילדים של צומת לפי label
7. החזר label של ילד לפי label קשת וlabel הורה
8. החזר label הורה לפי label קשת ו- label ילד

בשביל למנוע שכפול קוד, החלטנו להוסיף את המתודות הפנימיות הבאות:

1. הוספת צומת (מקבלת label וצבע)

כמו כן, הוספנו מחלקה בשם ColoredVertex, אשר מממשת את ההפשטה של צומת. למחלקה זו הפעולות הבאות:

1. הוסף הורה לפי label
2. הוסף ילד לפי label
3. האם הורה קיים לפי label
4. האם ילד קיים לפי label
5. האם קשת נכנסת קיימת לפי label
6. האם קשת יוצאת קיימת לפי label
7. האם הצומת שחור
8. האם הצומת לבן
9. החזר רשימת labels של הורים
10. החזר רשימת labels של ילדים
11. החזר ילד לפי label קשת
12. החזר הורה לפי label קשת

התלבטנו האם לממש גם מחלקה של Edge או להתייחס לכל קשת רק בתור ה-label שלה. בשל המניעים הבאים החלטנו לא לממש מחלקה ייעודית:

1. אין דרישה בהפשטה להחזיר את כל הקשתות של הגרף, או אף את כל הקשתות של צומת, ועל כן אין צורך לשמור מידע על כל הקשתות בצורה מרוכזת.
2. קשת מוגדרת חח"ע ע"י מוצא, יעד ו-label. ע"מ לשמור מוצא ויעד בקשת, ובמקביל לשמור את הקשתות המחברות כל צומת, עלינו לבצע ייבוא (import) לולאתי.
3. בהמשך לנימוק 2: אמנם ניתן לשמור במחלקה Edge רק את ה-label-ים של המוצא ושל היעד, אבל שמנו לב שלפי ההפשטה שמידע זה הינו redundant, מכיוון שבכל חיפוש של ילד או הורה לפי קשת מתקבל גם פרמטר של label הורה או label ילד (בהתאמה), כך שניתן באמצעות label צומת זה ו-label קשת למצוא בצורה יעילה את הצומת המבוקש גם ללא אחזקת הנתונים באובייקט מסוג Edge.

לדעתנו, פעולות אלו מספקות למימוש ההפשטה. הן מאפשרות בניית גרף ע"י הוספת צמתים (לפי צבעים) וקשתות, אחזור רשימת צמתים לפי צבע, אחזור הורים וילדים של צומת, אחזור צומת לפי הורה/ילד וקשת. פעולות אלו מאפשרות לקבל גם מידע כללי על הגרף, וגם בניית מסלולים ותתי- גרפים מתוך הגרף.

## סעיף ב' – תיעוד חיצוני

כפי שציינו בסעיף הקודם, בחרנו לממש את הגרף בעזרת מבנה נתונים נוסף של צומת.

לצורך סעיף זה נסמן ב- nאת מספר הצמתים בגרף.

**מימוש הצומת:**

1. T label - התווית של אותו הצומת.
2. HashMap<T, T> parents – טבלת ערבול מתווית של קשת לתווית של צומת. טבלת ערבול זו מכילה את כל הקשתות הנכנסות לאותו הצומת. מאפשר לגשת לכל אחד מהאבות של הצומת ב-O(1) משוערך.
3. HashMap<T, T> children – טבלת ערבול מתווית של קשת לתווית של צומת. טבלת ערבול זו מכילה את כל הקשתות היוצאות מאותו הצומת. מאפשר לגשת לכל אחד מהבנים של הצומת ב-O(1) משוערך.
4. VertexColor color – הצבע של אותו הצומת.

**מימוש הגרף:**

1. HashMap<T, ColoredVertex<T>> vertexes – טבלת ערבול מתווית לצומת, מכילה את כל הצמתים בגרף. מאפשר גישה ב-O(1) משוערך לצומת בגרף.
2. Set<T> blackVertexes – קבוצה של כל התוויות של הצמתים השחורים בגרף. מאפשר להחזיר את כל הצמתים השחורים בגרף ב-O(1).
3. Set<T> whiteVertexes – קבוצה של כל התוויות של הצמתים הלבנים בגרף. מאפשר להחזיר את הצמתים הלבנים בגרף ב-O(1).

הפעולות שנדרשות לביצוע בזמן מהיר הן:

1. מציאת רשימת בנים לצומת מסוים:

בעזרת מבני הנתונים שתחזקנו יהיה אפשר לקבל את רשימת הבנים של צומת מסויים ב-O(1) משוערך. הסיבוכיות במקרה הגרוע תהיה כאשר זה הדרגה של הצומת שנרצה לקבל את רשימת הבנים שלו.

1. מציאת רשימת אבות לצומת מסוים:
2. בעזרת מבני הנתונים שתחזקנו יהיה אפשר לקבל את רשימת האבות של צומת מסויים ב-O(1) משוערך. הסיבוכיות במקרה הגרוע תהיה כאשר זה הדרגה של הצומת שנרצה לקבל את רשימת האבות שלו.
3. בניית הגרף:
   1. הוספת צומת לגרף:

בהוספת צומת נצטרך ליצור מבנה נתונים מסוג צומת: יצירת שתי טבלאות ערבול ריקות – O(1). בנוסף נצטרף להוסיף התווית של הצומת לקבוצה המתאימה – O(1) ולהוסיף את הצומת לטבלת הערבול של הצמתים – O(1) משוערך ו-O(n) במקרה הגרוע.

בסכך הסיבוכיות של הפעולה: O(1) משוערך, O(n) במקרה הגרוע.

* 1. הוספת קשת לגרף:
     + גישה לשני הצמתים שהיא מחברת – O(1) משוערך, O(n) במקרה הגרוע.
     + בדיקה שאין קשת עם תווית זהה שנכנסת או יוצאת לאחד מהצמתים האלו, לשם כך, נבדוק ברשימת האבות או הבנים של כל אחד מהצמתים – O(1) משוערך, O(n) במקרה הגרוע.
     + בדיקה ששני הצמתים לא מאותו צבע – O(1).
     + הוספת הקשת לרשימת האבות והבנים המתאימות - O(1) משוערך, O(n) במקרה הגרוע.

בסך הכל:

O(1) משוערך, O(n) במקרה הגרוע.

קיבלנו שכל הפעולות שהוגדרו כקריטיות יבוצעו בסיבוכיות משוערכת O(1).

## סעיף ג' – מימוש חלופי

במימוש שלנו בחרנו להשתמש ב-HashMap לשם שמירת מידע במבני הנתונים (שמירת האבות והבנים בכל צומת ושמירת הצמתים בגרף). באותה מידה יכולנו לבחור להשתמש ברשימות מקושרות. אם היינו בוחרים לממש כך את מבני הנתונים סיבוכיות הזמן המשוערכת לפעולות: גישה לצומת או גישה לבן או אב של צומת הייתה גדלה. במימוש שבחרנו סיבוכיות הזמן המשוערכת לפעולות אלו היא O(1) ואילו עבור רשימה מקושרת היא O(n) כאשר n מספר האיברים ברשימה.

לעומת זאת, סיבוכיות הזמן במקרה הגרוע לא תשתנה גם אם נבחר להשתמש ברשימות מקושרות ותישאר לינארית למספר האיברים.