תרגיל 9- מבני נתונים 67109:

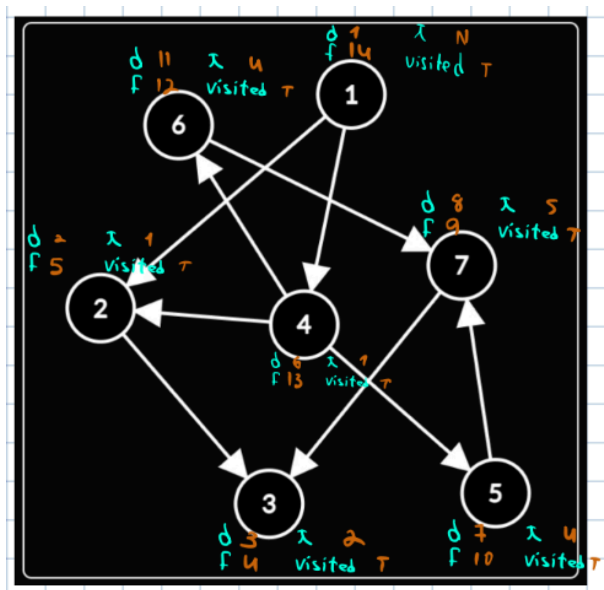
שם: דן קורנפלד

חלק א:

1. שאלה 1:

(a) נריץ DFS על הגרף, ונחזיר רשימה $v_1,...,v_n$ כך ש $f(v_1) > ... > f(v_n)$

(b) לכן מיון טופולוגי אופציונלי יהיה: $1, 4, 6, 5, 7, 2, 3$



1. שאלה 2:

(a) ראשית נתונה לנו רשימה של רשימות, כך שבכל רשימה נתון דירוג של משחק כלשהו. נבנה גרף מכוון באמצעות רשימת שכנויות, כך שכל רשימה תהיה טבלת גיבוב (שגדלה באופן לינארי כגודל מספר השכנויות, לדוגמה: עבור כל שכן, גודל רשימת השכנויות תהיה בגודל $4n$, כדי שטבלת הגיבוב לא תהיה צפופה מידי, ותעבוד כראוי), כך שכל קודקוד יהווה שחקן במשחק.

(b) נעבור על כל רשימת דירוג של משחק ונוסיף צלעות לפי הדירוג: עבור שחקן x שבדיוק מעל השחקן y, נוסיף צלע שיוצאת מ x ונכנסת אל y (כאמור x מעל y). נבצע זאת עבור כל רשימות הדירוגים, נבדוק כמובן לפני שאם יש כבר צלע קיימת, לא נוסיף צלע נוספת).

(c) במצב זה יש לנו גרף מכוון ונרצה בהמשך להשתמש במיון טופולוגי כדי לקבל דירוג של שחקנים תקין, אך כדי שנוכל להשתמש במיון הטיפולוגי, נצטרך לבדוק שאין לנו מעגלים בגרף $\Longleftrightarrow$ כל רכיב קשירות חזק הוא קודקוד בפני עצמו.

(d) כדי לוודא שכל קודקוד הוא רכיב קשירות חזק בפני עצמו, נריץ את האלגוריתם של SCC, כך נוכל לבדוק בריצת ה-DFS (האחרונה, על הגרף המשוחלף) כך שנבדק לכל קודקוד באיזה רכיב קשירות הוא נמצא, ונוודא שכל אחד נמצא ברכיב קשירות עצמאי.

(e) במידה וקיים מעגל (כלומר יש רכיב קשירות חזק שמכיל יותר מקודקוד אחד) לא ניתן להגדיר דירוג אקראי ונחזיר false.

(f) לכן אם לא קיים מעגלים, נוכל לסדר את הדירוג במיון טיפולוגי, מכיוון שזהו סידור של קבוצת איברים שקיימות בניהם תלויות כך שאף איבר לא יופיע לפני איבר בו הוא תלוי (בדיוק הגדרת הבעיה שבה אנו נמצאים), ולכן נקבל דירוג שחקנים תקין.

(g) סיבוכיות זמן ריצה:

i. הגדרת רשימת השכנויות $O(n)$.

ii. יש לכל היותר $m \cdot k$ צלעות, מכיוון שיש בכל משחק k צלעות (כל שחקן מצביע כל השחקן הבא), ו m משחקים, לכן $m \cdot k$.

iii. עבור מעבר על רשימת הרשימות: $O(m \cdot k)$ (כלומר מעבר על כל התאים ברשימה של הרשימות- כל השחקנים בכל המשחקים). בניית הגרף- מכיוון שאנחנו משתמשים ברשימת שכנויות עם טבלת גיבוב, נוכל לבדוק ב O(1) בתוחלת האם קיים צלע שמחברת בין 2 קודקודים.

iv. ביצוע אלגוריתם SCC: $O(|V|+|E|)$ כלומר $|V|=n$, $|E|=m \cdot k$ לכן $O(n+m \cdot k)$

v. מיון טופולוגי: $O(|V|+|E|)$ ולכן לפי הנתונים: $O(n+m \cdot k)$ (ניתן להשתמש בסיבוכיות זמן של שאלה 4 שחוסכת את המיון הנוסף של הרשימה)

vi. לכן סיבוכיות זמן הריצה תהיה $O(n+m \cdot k)$

תרגיל 9- מבני נתונים 67109:

שם: דן קורנפלד

חלק 2:

שאלה 3:

עבור השאלה כיצד למצוא את המשקל המינימלי של קודקוד u שניתן להגיע מ-v, אציע את האלגוריתם הבא:

1. תחילה נעבור על כל הגרף בעזרת האלגוריתם שלמדנו של SCC ($strongly\ connected\ component$), כלומר נרצה למצוא את הרכיבי הקשירות החזק הגדולים ביותר בגרף. מצב זה יאפשר לנו להשקיע אמנם זמן, אך יחסוך בהמשך זמן בכך שהקודקוד המינימלי יהיה שווה בין כל רכיב הקשירות החזקה.
2. ברגע שאנחנו סורקים את הG^T ועוברים על כל רכיב קשירות, נשמור אותו בגרף G_{SCC} שיפורט בהמשך, ונבדוק תוך כדי הסריקה עבור הערך המינימלי של הרכיב קשירות (כדי שלאחר מכן נוכל להשוואת מול רכיבי הקשירות האחרים, יוכל לחסוך בזמן), וגם את הערך נשמור במבנה הנתונים שיחזיק את ה G_{SCC}.
3. לאחר שביצענו את ה SCC נבנה את ה G_{SCC} כך שכל קבוצה של רכיב קשירות חזקה יהווה קודקוד אחד.
4. הבניה באופן הבא: נעבור על כל קודקוד ונעבור על כל הקודקוד ברשימת השכנויות של אותו קודקוד, נבדוק האם קיים כבר צלע בגרף החדש, אם לא- נוסיף את הצלע, אם כבר יש צלע- לא נוסיף צלע נוספת. כך למעשה נעבור על כל הצלעות כדי לבנות את הגרף החדש. נוכל לייעל את זמן הבדיקה האם כבר יש צלע בגרף החדש, כשנבנה את ה G_{SCC} כך שהקודקוד יכיל טבלת גיבוב לקודקודים של רכיב הקשירות, וטבלת גיבוב עבור רשימת השכנויו.
5. כלומר: נשמור את הגרף G_{SCC} בטבלת גיבוב (במקום רשימה מקושרת רגילה) כך שכל תא יכיל טבלת גיבוב של כל רכיבי הקשירות חזקה, את המשקל המינימלי של כל הרכיב קשירות ורשימת שכנויות שאליה הרכיב קשירות מחובר (הרשימה מכילה את לאילו רכיבי קשירות החצים מגיעים, כנלמד בקורס, נבצע זאת כאמור כ**טבלת גיבוב**).
6. לאחר מכן, נבצע מיון טופולוגי (ביצוע DFS על הגרף), ומכיוון שאין לנו מעגלים, תחילה אנחנו נעבור מקודקוד מסוים "לעומק" הגרף עד שנגיע לקודקוד האחרון, הקודקוד האחרון יכיל את הערך המשקלי שלו עצמו, ויחזיר רקורסיבית את הערך המשקלי שלו, לקודקוד לפניו. הקודקוד לפניו יסתכל על המינימום בין: הרכיב שהיה לפניו, לבין הרכיב הנוכחי וישמור את המינימום בין ה-2, כך המיון ימשיך עד שנגיע למצב שכל קודקוד מעודכן על ידי המשקל המינימלי השייך לו. כך נבצע עבור כל רכיבי הקשירות (בכל פעם נבדוק את הערך המינימלי בין הנוכחי לבין הרכיבים שהוא מחובר אליהם).
7. לבסוף נקבל שלכל קודקוד יש ערך נוסף עם המשקל המינימלי, כך שיש מסלול בין הקודקוד לקודקוד אחר עם משקל מינימלי.
8. סיבוכיות זמן ריצה: ביצוע אלגוריתם + בניית הגרף G_{SCC} סיבוכיות זמן ריצה של $O(|V|+|E|)$ (בתוחלת), ביצוע מיון טופולוגי נוסף כדי לעדכן את ערכי המינימום $O(|V|+|E|)$, לכן סיבוכיות זמן הריצה של כל הפונקציה היא $O(|V|+|E|)$ (בתוחלת).

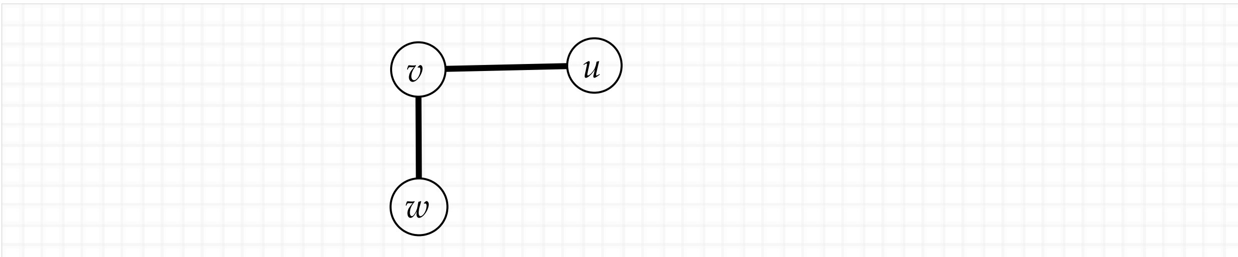
תרגיל 9- מבני נתונים 67109:

שם: דן קורנפלד

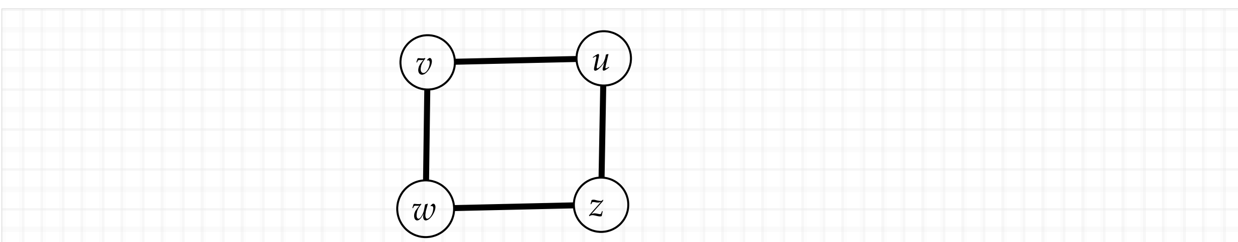
חלק 3:

שאלה 4:

1. סעיף 1: הפרכה: נראה שעבור הצלע שמחברת בין קודקוד u לקודקוד v, אם ננתק אותה אכן יווצרו 2 רכיבי קשירות (u בנפרד, v,w בנפרד), אך אם ננתק את u (וכתוצאה מזה גם את הצלע u,v) אזי לא נקבל 2 רכיבי קשירות, ולכן u לא תהיה נקודת מפרק.



2. סעיף 2: הפרכה: ניקח את הגרף המתואר מטה, נבנה עץ BFS ונראה שהשורש יהיה v, הבנים שלו הם u,w ובהתאם לסריקה, z יהיה הבן של u או w (לצורך הדוגמה, יהיה בן של u) על פניו, ל-v יש לפחות 2 בנים, עדיין הוצאה שלו לא גורמת לפיצול של רכיבי הקשירות ולכן לא יתקיים שהוא נקודת מפרק.



3. סעיף 3: הוכחה: בתחילת אלגוריתם הDFS ניצור רשימה ריקה בגודל $|V|$, בכל שלב שאלגוריתם יוצא מקודקוד כלשהו, הוא מתחיל בהתחלה מהערך הקטן ביותר ולאחר מכן הערך גדל לכן בכל שלב נכניס את האיברים מהסוף להתחלה, כך שהאיבר הראשון יהיה במקום האחרון וכך נכניס את כולם באופן ממוין עד האיבר עם ערך היציאה הגדול ביותר יהיה הראשון ברשימה הריקה.
   (a) ניתוח זמן ריצה: יצירת המערך בסיבוכיות זמן ריצה של $O(1)$, גישה למערך והשמת מספר בסיבוכיות זמן ריצה של $O(1)$, ביצוע DFS בסיבוכיות זמן של $O(|V|+|E|)$ ולכן זה סיבוכיות הזמן של האלגוריתם כמבוקש.