

חישוב ביולוגי – שב 2

מגשים:

נטע גלזר – 313372666

נועם דיאמנט – 208520262

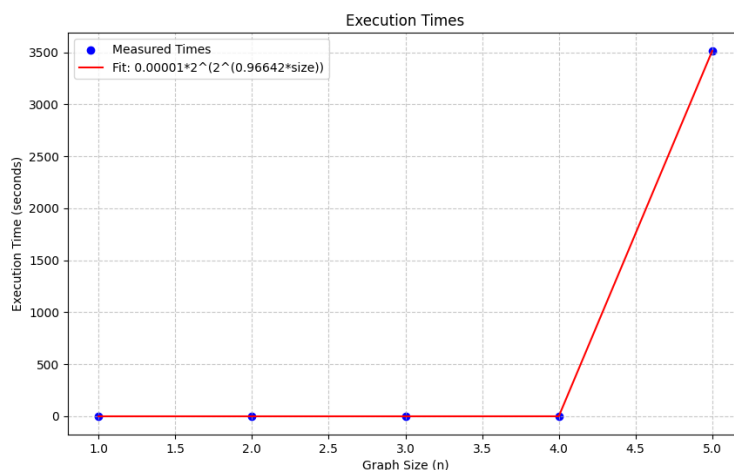
חלק 1

(a) תהליך מימוש הקוד כלל שני שלבים עיקריים:

1. פיתוח מתודה למציאת כל הגרפים הקשירים בעלי מספר נתון של קודקודים.
 2. סינון הגרפים הקשירים שהתקבלו כך שנשמר רק נציג יחיד מכל קבוצת גרפים איזומורפיים (כלומר, גרפים שניתן להגיע מאחד לשני על ידי שינוי שמות הקודקודים). הסינון הזה הכרחי מכיוון שבייצוג הפנימי של הקוד, לשמות הקודקודים יש משמעות טכנית (לצורך שמירת הגרף בזיכרון), אך מבחינת הבעיה המוצגת בשאלה, שמות הקודקודים אינם רלוונטיים.
- לצורך יעילות המימוש, נעשה שימוש בפונקציות `permutations` ו-`combinations` מהספרייה `itertools`, המאפשרות חישוב מהיר של קומבינציות אפשריות של קשתות ופרמוטציות של קודקודים בגרף.
- (b) כדי לקבל את התוצאה הרצויה ניתן להריץ את קובץ `part1.py` לפי ההוראות שמפורטות בגיטהאב של המטלה.

(c) סיבוכיות הזמן של הקוד היא מאוד גדולה, וזאת משתי סיבות עיקריות:

1. חישוב כל הקומבינציות האפשריות של קשתות לצורך יצירת הגרפים הקשירים
 2. בדיקת כל הפרמוטציות האפשריות עבור כל גרף קשיר (במקרה הגרוע)
- מבחינת ביצועים בפועל, התוכנית מתנהגת כך: עבור גרפים בגודל 4 ומטה קיבלנו זמן ריצה של פחות משעה, ועבור גרפים בגודל 5 קיבלנו זמן ריצה של כשעה.
- (d) כפי שציינו, התוכנית מסוגלת לבצע חישובים עבור גרפים עד גודל 5 בזמן סביר (פחות משעתיים). עם זאת, עבור גרפים בגודל 6 ומעלה, זמן הריצה הופך לבלתי מעשי, עם ריצות הנמשכות מעל 8 שעות, וזאת בשל הסיבוכיות החישובית שהוסברה. ניתוח זמני הריצה, המוצג בגרף, מדגים גידול אקספוננציאלי (ניתן לראות זאת בשערוך האקספוננציאלי שהוספנו) חד במיוחד: עבור גודל 4 קיבלנו זמן ריצה של פחות משנייה, ועבור גודל 5 - זמן ריצה של כשעה. פער זה ממחיש את הגידול הדרמטי בזמני הריצה, ומוביל למסקנה כי גודל 5 הוא הגודל המקסימלי המעשי עבור שני הסעיפים בתרגיל.



חלק 2

מימוש הפונקציה למציאת מוטיבים מתבסס על הפונקציונליות שפותחה בחלק הקודם, ופועל בשני שלבים:

1. יצירת כל המוטיבים האפשריים בגודל נתון באמצעות המתודה מהחלק הקודם.

2. חישוב מספר ההופעות של כל מוטיב בגרף הנתון.

לצורך זיהוי וספירת ההופעות של כל מוטיב, נדרשת בדיקה של כל פרמוטציות הגרף (בכל אחת מהן צריך לבדוק האם המוטיב מופיע באותה פרמוטציה). הסיבה לכך היא שמבחינת הבעיה, סדר ושמות הקודקודים אינם משמעותיים, אך בייצוג הפנימי של הקוד הם כן. מכיוון שאיננו יודעים מראש היכן בגרף נמצא המוטיב, עלינו לבדוק את כל הסידורים האפשריים של הקודקודים (אלו כל הפרמוטציות).

נשים לב שכאשר הגרף מכיל יותר קודקודים מהמוטיב, כל מוטיב (אם קיים) יופיע $(|V_G| - |V_M|)!$ פעמים, כאשר:

V_G הם קודקודי הגרף.

V_M הם קודקודי המוטיב.

הסיבה לכך היא שכאשר אנו "מקבעים" קודקודים בגרף להתאים לקודקודי המוטיב (כלומר, בפרמוטציה מסוימת של הגרף בה שמות קודקודי הגרף תואמים לקודקודי המוטיב), שאר הקודקודים (שאינם חלק מהמוטיב) יכולים להיות מסודרים בכל סדר אפשרי. כל סידור כזה מהווה פרמוטציה חוקית שבה המוטיב מופיע (מפני שכאמור, קיבענו את קודקודי הגרף שמהווים את המוטיב). חשוב לשים לב לעובדה שמספר הסידורים האפשריים של הקודקודים הנותרים הוא העצרת של מספרם. לכן, לאחר ספירת כל ההופעות, יש לחלק את התוצאה בערך זה כדי לקבל את מספר ההופעות האמיתי של המוטיב.

קישור לגיטהאב:

<https://github.com/Noam-Diamant/BioComp/tree/main>