

מטלה 6 אלגוריתמים כלכליים:

שאלה 4 א' תיעוד לפונקציות:

לפני הפונקציה שעוברת על הגרף ומחפשת מעגלים –findCycleVal, כתבתי שני פונקציות שעוזרות ביצירת הגרף

- חשוב לציין כי הגרף הינו גרף מכוון בעל משקלים על הצלעות.

הפונקציה הראשונה :createGraphNodes

מקבלת מספר קודקודים להוסיף לגרף ומוסיפה אותו על ידי פעולה של networkx. השתמשתי ב graph.add_nodes_from(H), כאשר H מכיל את הקודקודים

פונקציה שניה : createGraphEdges

פונקציה המקבלת גרף ורשימה של צלעות כך שכל אובייקט ברשימה הוא רשימה פנימית המכילה קודקוד התחלה של הצלע, קודקוד סיום והמשקל של הצלע

הפונקציה עוברת על האובייקטים ברשימה ומוסיפה על ידי פעולה של network, את הצלעות לגרף.

להוספה השתמשתי ב graph.add_edge.

הפונקציה הראשית בקוד הינה –findCycleVal, מקבלת גרף ובודקת האם יש מעגל בגרף, כך שמכפלת ערכי צלעותיו קטנה מאחד.

תחילה הפונקציה מקבלת את כל המעגלים האפשריים בגרף מכוון לאחר מכן הפונקציה עוברת על כל מעגל ומעגל

הפונקציה בודקת האם מכפלת כל ערכי הצלעות שבמעגל הנוכחי קטנה מאחד – אם כן מחזירה את המעגל הנוכחי אחרת אם עברנו על כל המעגלים

ולא נמצא מעגל עם מכפלה שערכה קטן מאחד נחזיר None

*לפירוט מורחב על הקוד ניתן להסתכל בהערות באנגלית ב github של המטלה

```
def findCycleVal(graph: nx.Graph):
    currentCycles = (nx.recursive_simple_cycles(graph))
    # go over all the possible cycles in the graph
    for currCycle in currentCycles:
        multiplyVal = 1
        i = 0
        prevNode = currCycle[i]
        i = i + 1
        # go over all the edges in the current cycle
        while i < len(currCycle):
            currNode = currCycle[i]
            # multiply the current amount with the edges value
            multiplyVal = multiplyVal * graph[prevNode][currNode]["weight"]
            prevNode = currNode
            i = i + 1
        i = currCycle[0]
        multiplyVal = multiplyVal * graph[prevNode][i]["weight"]
        if multiplyVal < 1:
            return currCycle
    return None
```

שאלה 4 ב':

* ב. האם אפשר לפתור את הבעיה בעזרת אלגוריתם דייקסטרה, עם החלפת סכום במכפלה? אם כן - כתבו קוד העושה זאת והוכיחו את נכונותו. אם לא - הוכיחו שזה בלתי אפשרי.

תחילה אסביר את אלגוריתם dijkstra,

אלגוריתם זה מתחיל לרוץ בגרף, מקודקוד התחלה מסוים. לפני כן הוא מכניס לכל הקודקודים ערך גדול מאוד (אינסוף)

כאשר נעבור מהקודקוד ההתחלתי נעדכן את הערך של הקודקודים השכנים שלו לפי הערך של הצלע שבניהם – כמובן שנבדוק לפני האם אנחנו מקטינים את הערך של הקודקוד אליו אנו עוברים ואם כן נעדכן את הערך של השכן.

במהלך האלגוריתם הקודקודים נשמרים בתור עדיפויות לפי הערך הכי קטן (הקטן יהיה בתחילת התור, עם עדיפות גבוהה יותר)

אם נשנה את שיטת עדכון הקודקודים, שהיא לסכום את הצלע יחד עם הערך של הקודקוד הנוכחי- במכפלה של הצלע יחד עם הערך של הקודקוד הנוכחי. בנוסף ניתקל במצב שישנו מעגל בגרף שמכפלתו קטנה מאחד.

נקבל שניכנס ללולאה אינסופית בגרף מכיוון שכל פעם שנעבור על המעגל הזה נקטין את הערך של הקודקוד הנוכחי כי נכפיל את הערך בשבר שקטן מאחד

תהליך זה שקול לשינוי שיטת עדכון הקודקודים, לשיטה הבאה: שהיא לסכום את הצלע יחד עם הערך של הקודקוד הנוכחי. אך הערכים יהיו ב(log), גם בצלע וגם בקודקוד הנוכחי. זאת מכיוון שלפי החוק הבא:

$$\text{Rule 2: } \log_b \left(\frac{M}{N} \right) = \log_b M - \log_b N$$

ומכאן כאשר יש נו מעגל בגרף שסכומו בשיטה הנוכחית קטן מ1
 נקבל ערכים שלילים על הצלעות וכך גם ניכנס ללולאה אינסופית
 לסיכום לא ניתן לפתור את הבעיה בעזרת אלגוריתם dijkstra, מכיוון שנוכל להיכנס ללולאה אין
 סופית בהרצה.