מטלה - חלוקת חפצים בדידים – פתרון

שאלה 1: חלוקה ללא קנאה מלבד החפץ הכי קטן

הגדרה: חלוקה של חפצים נקראת "ללא קנאה מלבד החפץ הכי קטן" (בקיצור EFX), אם לכל שני שחקנים א, ב, **ולכל** חפץ שנוריד מהסל של ב, שחקן א לא יקנא בשחקן ב.

א. הראו שהתנאי EFX הוא ממש חזק יותר מהתנאי EF1 (ע"י דוגמה לחלוקה EF1 שהיא לא EFX).

ב. יש שני שחקנים עם העדפות אדיטיביות (= לכל שחקן, ערך של סל הוא סכום הערכים של החפצים בסל), וזהות (= לשני השחקנים יש אותם ערכים לכל החפצים). תארו אלגוריתם המוצא חלוקה EFX.

ג. יש שני שחקנים עם העדפות אדיטיביות, אבל לא בהכרח זהות. תארו אלגוריתם המוצא חלוקה EFX.

ד. יש שלושה שחקנים עם העדפות אדיטיביות לא זהות. הוכיחו או הפריכו: תמיד קיימת חלוקה EFX (היעזרו ב Google Scholar).

- **פתרון**: אופיר

* שאלה 2: חלוקה כמעט-ללא-קנאה עם כמות זהה

נתונים n אנשים ו m*n חפצים. צריך לחלק את החפצים לאנשים כך ש:

- החלוקה היא ללא קנאה עד כדי חפץ אחד (EF1);
- כל אדם מקבל בדיוק m חפצים.

א. הראו (בעזרת דוגמה פשוטה) שאלגוריתם גרף הקנאה לא מתאים למשימה זו.

ב. הראו (בעזרת דוגמה פשוטה) שאלגוריתם מיקסום מכפלת הערכים לא מתאים למשימה זו.

- **פתרון**: נתנאל

ג. תארו אלגוריתם אחר המבצע את המשימה. הוכיחו את נכונות האלגוריתם שלכם.

- **פתרון**: https://en.wikipedia.org/wiki/Round-robin_item_allocation

שאלה 3: חלוקת תיקים בין מפלגות בקואליציה

לאחר הבחירות, שתי מפלגות החליטו להקים ממשלה ביחד, אבל הן בזבזו חודש שלם בויכוחים על איזה מפלגה תקבל איזה תיק. לאחר שהתייאשו מהויכוחים, הן פנו אליכם כדי שתעזרו להם להחליט.

א. הציעו אלגוריתם שיאפשר לשתי המפלגות להחליט על חלוקת-תיקים עם התכונות הבאות:

- יעילה פארטו.
- מותר שתהיה רוטציה בין המפלגות בתיק אחד לכל היותר.
- החלוקה תהיה הוגנת, בהתחשב בגדלים השונים של המפלגות.

הגדירו במדוייק את תכונת ההגינות, והוכיחו את נכונות האלגוריתם.

ב. הדגימו את פעולת האלגוריתם שלכם על אוסף התיקים שהיו באחת מממשלות ישראל האחרונות (לבחירתכם), ועל שתי מפלגות לבחירתכם.

- **פתרון**: תם

שאלה 4: חלוקה הוגנת ויעילה עם מספר שיתופים קטן ביותר

נתונים שני אנשים ו-m חפצים. יש למצוא חלוקה ללא-קנאה ויעילה-פארטו של החפצים, כך שמספר החפצים ה"נחתכים" (או נשארים בבעלות משותפת) הוא הקטן ביותר האפשרי.

א. הוכיחו שהבעיה היא NP-קשה (כפונקציה של m). רמז: https://en.wikipedia.org/wiki/Partition_problem.

פתרון:

- בעיית ה-Partition ידועה כבעייה NP-קשה. כדי להוכיח שגם בעיית שיתוף-מינימלי היא NP-קשה, צריך להראות רדוקציה – להראות, שאילו היה פתרון בזמן פולינומיאלי לבעיית שיתוף-מינימלי, אז היינו יכולים להשתמש בו כדי להשיג פתרון בזמן פולינומיאלי לבעיית ה-Partition.
- נניח שיש פתרון בזמן פולינומיאלי לבעיית שיתוף-מינימלי, ונניח שאנחנו מקבלים קלט לבעיית Partition. הקלט הזה כולל m מספרים שלמים. אנחנו בונים בעיית שיתוף-מינימלי שבה יש שני שחקנים, נניח עמי ותמי, עם ערכים שוים לכל החפצים – הערך של כל אחד מהשחקנים לחפץ i הוא בדיוק מספר i בקלט. עכשיו אנחנו מריצים את האלגוריתם (שהנחנו שקיים) על הבעיה הזאת, ובודקים את התוצאה. לפי ההנחה, התוצאה היא חלוקה ללא קנאה. כיוון שעמי ותמי מייחסים בדיוק אותו ערך לכל חפץ, חלוקה ללא קנאה חייבת לתת סכום-ערכים זהה לעמי ולתמי. עכשיו יש שתי אפשרויות:
    - אם בפלט של אלגוריתם "שיתוף מינימלי" אין שיתופים בכלל – אז התוצאה היא פתרון לבעיית ה Partition – חלוקה של המספרים בקלט לשתית קבוצות עם סכום זהה.
    - אם בפלט של אלגוריתם "שיתוף מינימלי" יש שיתוף אחד או יותר – אז לפי ההנחה שזה מספר השיתופים המינימלי, אפשר להסיק ש**לא קיימת** חלוקה בלי שיתופים בכלל (אחרת האלגוריתם היה מוצא אותה), ולכן אין פתרון לבעיית ה Partition.
- הוכחנו, שכל אלגוריתם לבעיית שיתוף-מינימלי יכול לפתור גם את בעיית Partition; לכן בעיית שיתוף-מינימלי היא NP-קשה.

ב. הוכיחו, שאם הערכים שהאנשים מייחסים לחפצים נבחרים באקראי (למשל כל ערך הוא מספר ממשי המתפלג אחיד בין 0 ל-1, כל ההגרלות בלתי-תלויות סטטיסטית), אז בהסתברות 1, הבעיה ניתנת לפתרון בזמן O(m log m).

פתרון: אם הערכים נבחרים באקראי, אז בהסתברות 1, כל יחסי-הערכים שונים. לכן יש דרך אחת ויחידה לסדר את החפצים לפי סדר עולה של יחס-הערכים (וזה לוקח זמן m log m כידוע). כפי שהסברנו בסוף ההרצאה הקודמת, כל חלוקה יעילה-פארטו חייבת לתת חפצים לשני האנשים בהתאם ליחס הערכים. לכן, יש רק m+1 חלוקות יעילות-פארטו ללא שיתופים כלל: או שנותנים את כל החפצים לעמי, או שנותנים m-

1 חפצים לעמי ו-1 לתמי, וכו'. מספיק לבדוק את כל החלוקות האלו ולראות אם אחת מהן ללא קנאה. אם כן – מחזירים אותה. אם לא – אז לא קיימת חלוקה יעילה והוגנת ללא שיתופים, כלומר מספר השיתופים המינימלי האפשרי הוא 1. בשלב זה אפשר להריץ את אלגוריתם "המנצח המתוקן" ולמצוא חלוקה הוגנת ויעילה עם שיתוף אחד.

* שאלה 5: תיכנות - יצירת גרף הקנאה

נתונה המחלקה הבאה:

```
class Agent:
     def item_value(item_index:int)→float: ...
```

המחלקה מייצגת שחקן המשתתף במשחק חלוקה הוגנת. יש בה פונקציה אחת המתארת את הערך שהשחקן מייחס לחפץ שהאינדקס שלו הוא item_index.

כיתבו פונקציה המייצרת את גרף הקנאה בחלוקה נתונה.

```
def envy_graph(agents:List[Agent], bundles:List[List[int]]): ...
```

הפרמטר agents הוא מערך בגודל n המייצג את השחקנים.

הפרמטר bundles הוא מערך באותו גודל - n - המייצג את החלוקה: bundle[i] הוא אוסף אינדקסי החפצים שמקבל שחקן i.

הפונקציה מחזירה גרף של הספריה networkx של פייתון.

הדגימו את הפונקציה שלכם על מספר קלטים מעניינים.

פתרון: עמית רן https://github.com/amitRan109/Economic-algorithms-task-5