

מטלה – אלגוריתמי החלפה

יש לענות על שאלה אחת לבחירתכם. שאלות רגילות מזכות בנקודה אחת. שאלות או סעיפים עם כוכבית מזכים בנקודה נוספת.

שאלה 1: תיכנות - חיפוש מעגלי החלפה

א. כיתבו פונקציה בפייתון המוצאת מעגל בגרף החלפת בתים, כאשר לכל השחקנים יש העדפות חזקות. אל תשתמשו ב `networkx` – השתמשו בתכונות המיוחדות של גרף החלפת הבתים כדי לממש אלגוריתם יעיל יותר באופן ישיר.

כותרת הפונקציה:

```
def find_trading_cycle(preferences: List[List[int]]): ...
```

הניחו שבתחילת הפונקציה, כל אדם i גר בבית i . הפרמטר `preferences` מתאר את ההעדפות של בעלי-הבתים (מניחים שההעדפות חזקות – אין אדישות). לכל i , הוקטור `preferences[i]` מתאר את סדר ההעדפות של אדם i . למשל, אם:

```
preferences[11] = [15, 13, 11, 7, 8, ...]
```

המשמעות היא, שאדם מספר 11 הכי רוצה את בית 15, אחריו את בית 13, אחריו את בית 11 (הבית הנוכחי שלו), וכו'.

הפונקציה מחזירה וקטור המתאר את המעגל, למשל אם מוחזר הוקטור:

```
[11, 15, 17, 11]
```

המשמעות היא, שאדם 11 מקבל את בית 15, אדם 15 מקבל את בית 17, ואדם 17 מקבל את בית 11.

ב. בעזרת הפונקציה של סעיף א, כיתבו פונקציה המבצעת את אלגוריתם מעגלי המסחר על אנשים עם יחסי העדפה חזקים. בדקו את תשובתכם בעזרת `doctest`.

שאלה 2: שיבוצים יציבים יחידים חזקים וחלשים

א. למדנו בהרצאה, שכאשר יחסי-ההעדפה חזקים, יש שיבוץ יציב אחד ויחיד. הראו שהמשפט אינו נכון כאשר יחסי-ההעדפה חלשים [הסבירו קודם במילים שלכם מה בדיוק צריך להראות].

* ב. למדנו בהרצאה, שאלגוריתם ההחלפה של סבן-סתורמן מחזיר תמיד שיבוץ יציב-חלש. הראו שהאלגוריתם לא תמיד מחזיר שיבוץ יציב-חזק [הסבירו קודם במילים שלכם מה בדיוק צריך להראות].

שאלה 3: אלגוריתם "המעגל הגדול ביותר"

הגדרה: אלגוריתם "המעגל הגדול ביותר" הוא אלגוריתם הפועל כמו אלגוריתם מעגלי המסחר, פרט לכך שהוא מוצא בכל פעם את המעגל הגדול ביותר האפשרי (הבעיה NP קשה, אבל לצורך השאלה נניח שיש לנו מחשב חזק מאד שמסוגל לפתור אותה). הוכיחו או הפריכו את הטענות הבאות לגבי אלגוריתם המעגל הגדול ביותר:

א. כשיחסי-ההעדפה חזקים (אין אדישות), אלגוריתם המעגל הגדול ביותר הוא יעיל-פארטו ומגלה-אמת.

* ב. כשיחסי-ההעדפה חלשים (יש אדישות), אלגוריתם המעגל הגדול ביותר הוא יעיל-פארטו ומגלה-אמת.

שאלה 4: תיכנות - רכיב-קשירות-חזק סופי

א. כיתבו פונקציה בפייתון הבונה גרף-העדפות עבור האלגוריתם להחלפת בתים עם אדישויות:

```
def build_graph(valuations: List[List[int]]): ...
```

הניחו שבתחילת הפונקציה, כל אדם i גר בבית i . הפרמטר `valuations` מתאר את הערכים של בעלי-הבתים. לכל i , הוקטור `valuations[i]` מתאר את הערך שאדם i מיחס לבתים. למשל, אם:

```
valuations[11] = [100, 200, 200, 300, 250, ...]
```

המשמעות היא, שאדם i הכי רוצה את הבית הרביעי (- ערך 300), אחריו את הבית החמישי (- ערך 250), אחריו את הבית השני והשלישי (- ערך 200; הוא אדיש ביניהם), וכו'.

ב. בעזרת סעיף א, כתבו קוד הבודק אם יש בגרף רכיב קשירות חזק סופי.

שאלה 5: אלגוריתם מעגלי-המסחר כמקרה פרטי

נניח שאנחנו מריצים את אלגוריתם סבן-סתורמן על בעיה שבה לאנשים יש העדפות חזקות. האם התוצאה תהיה זהה לאלגוריתם מעגלי-המסחר? הוכיחו והדגימו.