

מטלה – חלוקה עם שיתופים – הרבה שחקנים

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

יש להוכיח את נכונות התשובות. אפשר לכתוב אלגוריתם בפסאודו-קוד, אבל הוא צריך להיות מדויק ומפורט ברמה של שפת תכנות.

שאלה 1: כמה שיתופים לכל חפץ?

א. מחלקים ארבעה חפצים בין ארבעה שחקנים. כמו שלמדנו בהרצאה, תמיד קיימת חלוקה עם לכל היותר שלושה שיתופים. יש כמה אפשרויות לחלוקה כזאת:

1. חפץ א משותף בין כל ארבעת השחקנים.

2. חפץ א משותף בין שלושה שחקנים, וחפץ ב משותף בין שני שחקנים.

3. חפץ א משותף בין שני שחקנים, חפץ ב בין שני שחקנים, חפץ ג בין שני שחקנים.

לכל אחת מהאפשרויות, הראו דוגמה שבה האפשרות הזאת היא האפשרות היחידה הנותנת חלוקה פרופורציונלית עם שלושה שיתופים (לדוגמה, עבור סעיף 1: הראו דוגמה להערכות של השחקנים, שבה בכל חלוקה פרופורציונלית, חפץ א משותף בין כל ארבעת השחקנים).

* ב. מחלקים ארבעה חפצים בין **חמישה** שחקנים. כמו שלמדנו בהרצאה, תמיד קיימת חלוקה עם לכל היותר **ארבעה** שיתופים. יש כמה אפשרויות לחלוקה כזאת:

4. חפץ א משותף בין כל חמשת השחקנים.

5. חפץ א משותף בין ארבעה שחקנים, וחפץ ב משותף בין שני שחקנים.

6. חפץ א משותף בין שלושה שחקנים, וחפץ ב משותף בין שלושה שחקנים.

7. חפץ א משותף בין שלושה שחקנים, חפץ ב בין שני שחקנים, חפץ ג בין שני שחקנים.

8. חפץ א משותף בין שני שחקנים, חפץ ב בין שני שחקנים, חפץ ג בין שני שחקנים, חפץ ד בין שני שחקנים.

לכל אחת מהאפשרויות, הראו דוגמה שבה האפשרות הזאת היא האפשרות היחידה הנותנת חלוקה פרופורציונלית עם ארבעה שיתופים.

שאלה 2: חלוקה לקסימין-אגליטרית לשחקנים עם זכויות שונות

נתונים n שחקנים עם זכויות שונות. הזכות של כל שחקן i היא w_i , ופונקציית הערך של i היא V_i .

חשבו, עבור כל שחקן i , מספר ממשי x_i , כך שאם נכפיל את פונקציית הערך של i ב- x_i , אז כל חלוקה לקסימין-אגליטרית תהיה חלוקה פרופורציונלית בהתחשב בזכויות השונות (היזכרו בהגדרה של חלוקה פרופורציונלית עם זכויות שונות ממטלה 1). שימו לב: x_i תלוי ב- w_i וב- V_i .

הדגימו והוכיחו את תשובתכם.

שאלה 3: גרף הצריכה בפייתון

כתבו פונקציה בפייתון, המקבלת מטריצה המייצגת חלוקת חפצים לאנשים, בונה את גרף הצריכה, ובודקת אם יש בו מעגלים, ואם יש – מחזירה מעגל כלשהו:

```
def find_cycle_in_consumption_graph(allocation: List[List[float]]):
```

הקלט הוא מטריצה, שבה האיבר בשורה i ובעמודה j הוא החלק מחפץ j שמקבל שחקן i .

לצורך חישובים בגרפים, השתמשו בספריה **networkx**. בדקו והדגו את תשובתכם.

שאלה 4: חיפוש מעגלים עם מכפלה קטנה מ-1

א. נתון גרף מכוון עם משקלים חיוביים על הקשתות. הגרף מיוצג בפייתון במבנה `DiGraph` של הספרייה **networkx**. כתבו קוד בפייתון הבודק אם בגרף נתון יש מעגל מכוון עם מכפלה קטנה מ-1, ואם יש – מחזירה מעגל כזה. העזרו באלגוריתמים הנמצאים בספריה. בדקו והדגו את תשובתכם.

* 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.

* שאלה 5: מציאת שיפור פארטו חלש בגרף ההחלפות

כתבו פונקציה בפייתון, המקבלת מטריצה המייצגת חלוקת חפצים לאנשים, ומטריצה הערכות של אנשים לחפצים, ומעגל בגרף הצריכה. הפונקציה מחשבת שיפור פארטו חלש המבטל לפחות קשת אחת במעגל.

```
def find_weak_pareto_improvement(
    allocation: List[List[float]],
    valuations: List[List[float]],
    cycle: list
):
```

המעגל `cycle` הוא מהצורה: [שחקן, חפץ, שחקן, חפץ, ... , שחקן]. למשל: [0, 13, 2, 12, 1, 11, 0] הוא מעגל שבו שחקן 0 משתף את חפץ 11 עם שחקן 1, שחקן 1 משתף את חפץ 12 עם שחקן 2, וכו'.

הפלט צריך להיות החלוקה החדשה – לאחר השיפור.

בדקו והדגו את תשובתכם.