

אוניברסיטת בר-אילן
פקולטה: מדעי הטבע
מחלקה: מדעי המחשב

אלגוריתמים כלכליים - פתרון

שם המרצה: אראל סגל-הלוי
שנה _____ ה'תש"ף סמסטר _____ א _____ מועד _____ ב
תאריך בחינה: י"ב באדר ה'תש"ף 8/3/2020
משך הבחינה: 3 שעות

- בבחינה חמש שאלות. מותר לענות על כולן.
- סכום הנקודות בכל השאלות הוא 120. מי שיקבל מעל 120 – ציונו יהיה 100.
 - שאלה 5 היא שאלת רשות - מומלץ להשאיר אותה לסוף, ואפשר לקבל 100 גם בלעדיה.

יש לענות במחברת הבחינה.

חומר עזר מותר בשימוש: דף-נוסחאות אישי בגודל פוליו (A4). אפשר לכתוב משני צדי הדף.

הנחיות לפתרון שאלות תיכנות:

- מותר לכתוב קוד בשפת פייתון, Java או C++ לפי בחירתכם.
- מותר לכתוב גם בפסאודו-קוד, בתנאי שהקוד מדויק ומפורט באותה רמה של שפת-תיכנות.
- יש לתעד את הקוד ולהסביר היטב בעברית מה עושה כל שורה ואיך זה מתאים לאלגוריתם.
- אתם לא נבחנים על התחביר של השפה אלא על האלגוריתם. העיקר שאוכל להבין מה עשיתם.

הנחיות כלליות:

- יש להסביר כל תשובה בפירוט. ניקוד מלא יינתן רק על תשובה נכונה עם הסבר נכון.
- אם נראה לכם ששאלה כלשהי אינה מוגדרת עד הסוף (חסרות הנחות מסויימות), הניחו את ההנחות שנראות בעיניכם הגיוניות ביותר בהתאם לשאלה.

בהצלחה!!

שאלה 1 [30 נק']. חלוקה כמעט-ללא-קנאה כשהעדפות זהות

נתונים m חפצים בדידים שיש לחלק באופן הוגן. נתונים n אנשים עם העדפות זהות - כולם מעריכים את החפצים באותו אופן. נתונה פונקציה גלובלית המקבלת קבוצה של חפצים ומחזירה את הערך שלהם עבור כל המשתתפים:

```
def value(bundle: Set[Item]): ...
```

כל משתתף בחלוקה מיוצג ע"י מחלקה שבה שדה אחד והוא שם המשתתף:

```
class Agent:
```

```
    name: str
```

א [10 נק']. כיתבו בשפת פייתון, או בפסאודו-קוד מפורט ומדויק, אלגוריתם המוצא חלוקה ללא-קנאה עד-כדי-חפץ-אחד (EF1). כותרת הפונקציה:

```
def find_EF1(items: List[Item], agents: List[Agent]): ...
```

הפונקציה מקבלת כקלט רשימה של אנשים ושל חפצים, וכותבת את החלוקה למסך, למשל:

```
Ami gets items 0, 2, 4
```

```
Tami gets items 1, 3, 5
```

מותר לכם להוסיף למחלקות שדות לפי בחירתכם.

ב [10 נק']. הוכיחו את נכונות האלגוריתם.

ג [10 נק']. הדגימו את פעולת האלגוריתם של סעיף א על שלושה אנשים ושישה חפצים. הדגימו שני מקרים:

- מקרה אחד שבו עמי, תמי ורמי מקבלים שני חפצים כל אחד;
- מקרה שני שבו עמי מקבל ארבעה חפצים, ותמי ורמי חפץ אחד כל אחד.

פתרון:

א+ב. אפשר להשתמש באלגוריתם שקראנו לו "אלגוריתם מעגלי הקנאה". אבל אפשר לפשט אותו משמעותית, כי כאשר כל ההעדפות זהות - לא יכולים להיות מעגלי קנאה! תמיד יש מישור שאף אחד לא מקנא בו - זה שהסל שלו הוא בעל ערך גלובלי נמוך ביותר.

לצורך הפתרון, נוסיף למחלקה Agent שדה bundle מסוג "קבוצה", המחזיק את הסל הנוכחי של כל שחקן. נוסיף גם פונקציה המוצאת שחקן שאף אחד לא מקנא בו:

```
def find_unenvied_agent(agents: List[Agent]):  
    return min(agents, key = lambda agent: value(agent.bundle))  
# מחזיר שחקן שערך הסל הנוכחי שלו הוא מינימלי
```

```
def find_EF1(items: List[Item], agents: List[Agent]): ...
```

```

# נוסף למחלקה Agent שדה bundle מסוג "קבוצה", המחזיק את הסל הנוכחי של כל שחקן:
for agent in agents:
    agent.bundle = {}

# נעבור על החפצים לפי הסדר:
for item in items:
    # ניתן את החפץ הבא לשחקן שאף אחד לא מקנא בו:
    unenvied_agent = find_unenvied_agent(agents)
    unenvied_agent.bundle.add(item)

# נדפיס את החלוקה הסופית:
for agent in agents:
    print(agent.name+" gets items " + agent.bundle)

```

ג.

- נניח שיש שישה חפצים שהערך של כל אחד מהם הוא 1. אז בשלושת הסיבובים הראשונים, הערך המינימלי לשחקן יהיה 0, ולכן כל שחקן יקבל חפץ; ובשלושת הסיבובים הבאים, הערך המינימלי לשחקן יהיה 1, ולכן כל שחקן יקבל עוד חפץ.
- נניח שיש שני חפצים שהערך שלהם 9, ועוד ארבעה חפצים שהערך שלהם 1. בשני הסיבובים הראשונים, החפצים שערכם 9 ניתנים לשני שחקנים כלשהם (שיש להם ערך 0) – נניח תמי ורמי. מכאן והלאה, בכל הסיבובים הבאים, השחקן עם הערך המינימלי הוא עמי, ולכן הוא יקבל את 4 החפצים הנותרים.

שאלה 2 [20 נק']: בחירת פרסומות לתחנת רדיו

אתם מנהלים תחנת רדיו שבה, בכל יום, מוקצבות בסה"כ 999 שניות לפירסומות.

לתחנה פונים n מפרסמים. לכל מפרסם i יש פרסומת באורך נתון וידוע x_i . אם הפרסומת של מפרסם i משודרת ברדיו, המפרסם מקבל ערך v_i ; הערכים האלה הם מידע פרטי של המפרסמים.

המטרה שלכם, כמנהלי התחנה, היא למקסם את סכום הערכים של המפרסמים, תחת האילוץ שסכום האורכים של הפרסומות הוא לכל היותר 999. במקביל, חשוב לכם לעודד את המפרסמים לגלות את הערכים האמיתיים שלהם לפירסומות.

א [10 נק']. נתון שמספר המפרסמים $n=10$. באיזה אלגוריתם תשתמשו? מדוע? הדגימו את האלגוריתם על הנתונים הבאים:

	מפרסם א	מפרסם ב	מפרסם ג	מפרסם ד
אורך הפרסומת בשניות	100	200	400	900
הערך למפרסם באלפי ש"ח	3	5	7	9

ב [10 נק']. נתון שמספר המפרסמים $n=500$. באיזה אלגוריתם תשתמשו? מדוע? הדגימו את האלגוריתם על הנתונים של סעיף א.

פתרון: כדי למקסם את סכום הערכים יש לפתור את בעיית התרמיל.

א. כיוון שמספר המפרסמים קטן, אפשר לפתור את בעיית התרמיל באופן מדויק ע"י בדיקת כל הצירופים – יש רק 1024 צירופים אפשריים. לכן אפשר להשתמש באלגוריתם VCG, שהוא אלגוריתם אמיתי המסתמך על מיקסום סכום הערכים. בדוגמה הנתונה, סכום הערכים המקסימלי הוא $a+b+g=15$, ולכן נבחר את מפרסמים א, ב, ג. נחשב כמה כל אחד משלם:

- בלי א, נבחר את $a+b$ (סכום = 12); עם א, הסכום של האחרים הוא $a+b+g=12$. לכן א משלם 0.
- בלי ב, נבחר את $a+g$ (סכום = 10); עם ב, הסכום של האחרים הוא $a+g+g=10$. לכן ב משלם 0.
- בלי ג, נבחר את ד (סכום = 9); עם ג, הסכום של האחרים הוא $a+b=8$. לכן ג משלם 1.

ב. כיוון שמספר המפרסמים גדול, ובעיית התרמיל היא NP-קשה, אי אפשר לפתור אותה באופן מדויק ע"י בדיקת כל הצירופים – 2 בחזקת 500 זה יותר מדי. לכן נשתמש באלגוריתם קירוב-חצי שלמדנו בהרצאה, וכדי שיהיה אמיתי נהפוך אותו למכרז בשיטת מאירסון.

- **אלגוריתם חמדני א** מסדר את המפרסמים בסדר: ד-ג-ב-א (מהערך הגבוה לנמוך). הוא בוחר את ד בלבד, סכום=9.

- **אלגוריתם חמדני ב** מסדר את המפרסמים בסדר: א-ב-ג-ד (מהיחס ערך/אורך הגבוה לנמוך. היחס עבור א הוא $3/100$; מכאן והלאה האורך גדל פי 2 או יותר, והערך גדל פחות מפי 2, ולכן ברור שהיחס יורד – לא צריך מחשבון!). הוא בוחר את $a+b+g$, סכום $=15$.

התוצאה של חמדני-ב גבוהה יותר ולכן נבחרים מפרסמים $a+b+g$. נחשב את ערך-הסף של כל אחד – עד לאן הם יכולים להוריד את הערך שלהם ועדיין להיבחר.

ראשית, ברור שאם כל אחד מהם יוריד את הערך שלו, חמדני-א לא ישתנה – הוא עדיין יבחר את ד.

שנית, חמדני-ב עדיין ישים את שני האחרים מהקבוצה $\{a, b, g\}$ לפני ד, ולכן גם חמדני ב עדיין לא יבחר את ד – הוא עדיין יבחר את $a+b+g$ [בהנחה שהאלגוריתם ממשיך לרוץ על כל השורה].

לכן, השינוי האפשרי היחיד הוא שהערך של $a+b+g$ יהיה נמוך מהערך של ד, ואז אלגוריתם הקירוב יבחר את התוצאה של חמדני-א. לכן:

- **ערך הסף של א** הוא 0, כי גם אם הערך שלו יהיה 0 – עדיין ייבחרו $b+g$.
 - **ערך הסף של ב** הוא 0, כי גם אם הערך שלו יהיה 0 – עדיין ייבחרו $a+g$.
 - **ערך הסף של ג** הוא 1 – כי אם הערך שלו יירד מתחת ל-1, הסכום של $a+b+g$ יהיה קטן מ-9 וייבחר ד.
- לסיכום, התשלומים במקרה זה זהים לתשלומים של VCG.

הערה: אם אלגוריתם חמדני-ב לא ממשיך לרוץ על כל השורה, אז ערכי הסף יהיו שונים – כי האלגוריתם יעצור ברגע שיגיע ל-ד (יחס ערך/אורך $=1/100$). לכן ערכי-הסף של א, ב, ג יהיו 1, 2, 4 בהתאמה.

שאלה 3 [30 נק': החלפת משמרות

א+ב [20 נק']. כיתבו פונקציה בפייתון (או פסאודו-קוד מדוייק ומפורט), המבצעת החלפת משמרות בין עובדים, בעזרת אלגוריתם מעגלי המסחר. העזרו במחלקה הבאה:

```
class Worker:
    name: str
    # Worker name, for display.
    current_shift: int
    # The shift to which the worker is currently assigned
    preferences: list
    # preferences[0] is the best shift for the worker.
    # preferences[1] is the 2nd-best shift for the worker. etc...
```

הפונקציה מקבלת כקלט וקטור עם כל העובדים במערכת (לכל עובד רשום סדר העדיפויות שלו והמשמרת הנוכחית שלו), וכותבת למסך את השינויים שיש לבצע בלוח השיבוצים, למשל:

```
Avraham moves from shift 0 to shift 1.
Isaac moves from shift 1 to shift 2.
Yaakov moves from shift 2 to shift 0.
```

כותרת הפונקציה:

```
def exchange_shifts (workers: List[Worker])
```

ג [10 נק']. הדגימו את פעולת הפונקציה על הקלט הבא:

```
a = Worker("Alice", 1, [3, 2, 4, 1])
b = Worker("Bob", 2, [3, 5, 6, 1])
c = Worker("Chloe", 3, [3, 1, 2, 4])
d = Worker("David", 4, [2, 5, 6, 4])
e = Worker("Edward", 5, [1, 3, 2])
f = Worker("Finn", 6, [2, 4, 5, 6])
workers = [a, b, c, d, e, f]
```

פתרון: ראו בפתרון של טל למטלה 11:

<https://github.com/SlrK/Economic-Algorithms/blob/master/11-3.py>

המשמרות הסופיות הן, לפי הסדר: 2, 5, 3, 6, 1, 4

שאלה 4 [20 נק']: חלוקת פנסי-רחוב בין שכונות

בעיר "אלגוריתמיה" ישנם 1200 תושבים המתגוררים בשלוש שכונות.

מספר התושבים בכל שכונה הוא: א – 546, ב – 504, ג – 150.

העירייה מעוניינת להקים פנסי-רחוב ומקציבה לשם כך 120 אלף ש"ח. העלות של כל פנס היא אלף ש"ח.

א [10 נק']. לפי אלגוריתם עזיז-לי-טלמון, כמה פנסי-רחוב יוצבו בכל שכונה? פרטו את שלבי הרצת האלגוריתם.

ב [10 נק']. לפי חוק באדר-עופר, כמה פנסי-רחוב יוצבו בכל שכונה? נמקו ופרטו את שלבי החישוב.

פתרון: [מסתמך על הפתרון של אייל ושלו למטלה 9 – תודה!]

א. יש כאן בעיה של תקצוב השתתפותי עם 360 פריטים – 120 בכל שכונה. צריך לסדר את כל תת-הקבוצות של פריטים בסדר יורד של עלות; במקרה זה העלות שווה לגודל. אפשר לדלג על קבוצות "מעורבות" כי אף אחד לא יתמוך בהן. בסה"כ אנחנו מקבלים רשימה מצומצמת של תת-קבוצות רלבנטיות:

[120 פנסים בשכונה א], [120 פנסים בשכונה ב], [120 פנסים בשכונה ג], [119 פנסים בשכונה א], [119 פנסים בשכונה ב], [119 פנסים בשכונה ג], [1 פנס בשכונה א], [1 פנס בשכונה ב], [1 פנס בשכונה ג], [קבוצה ריקה].

עוברים על התת-קבוצות מהגדולה לקטנה, ועוצרים ברגע שיש קבוצה שמספר התומכים בה מספיק גדול כדי לתקצב אותה. במקרה זה, כדי לתקצב קבוצה בגודל k צריך $10k$ תושבים. לכן נעצור על הקבוצה [54 פנסים בשכונה א] ונתקצב אותה. עכשיו, כל האזרחים בשכונה א יוצאים מקבוצת ה"מקופחים".

נמשיך כך ונעצור בפעם הבאה בקבוצה [50 פנסים בשכונה ב], ובאותו אופן [15 פנסים בשכונה ג].

בסה"כ יש 119 פנסים: $15 + 50 + 54$. נשאר עודף 1000 ש"ח בקופה.

ב. חוק באדר-עופר מוכר לנו מהבחירות לכנסת, אולם למעשה זה אלגוריתם כללי שמטרתו להחליט איך לנצל את עודף התקציב של 1000 ש"ח. הרעיון הוא, שבמקום 10 תושבים לכל פנס, אנחנו מורידים את ה"מחיר" לכל פנס עד שמספר הפנסים המתוקצבים מגיע ל-120. במקרה זה, אם נוריד את המחיר ל-9.927 נגלה שאפשר לתקצב את הקבוצה [55 פנסים בשכונה א], כיוון ש: $546/55 = 9.927$. התקציב לקבוצות האחרות לא ישתנה.

בסה"כ יהיו 120 פנסים: $15 + 50 + 55$, וכל התקציב מנוצל.

שאלה 5 [20 נק']. חישוב יתרה בשרשרת בלוקים

המטבע SimpleCoin הוא מטבע חדש (דמיוני), הדומה לביטקוין אבל הרבה יותר פשוט:

- בכל בלוק בשרשרת יש רק עיסקה אחת;
 - הסכום של כל עיסקה הוא מטבע אחד בדיוק;
 - לכל עיסקה יש רק נִמְעָן אחד (לא מפצלים מטבע לכמה נמענים שונים).
- ישנם שני סוגי בלוקים: בלוק שבו נוצר מטבע חדש (למשל כשכר לכוֹרֶה), ובלוק שבו מועבר מטבע שנוצר קודם. כל בלוק בשרשרת מיוצג ע"י המחלקה הבאה:

```
class Block:
    previous: Block
    # קישור לבלוק הקודם בשרשרת. אם הבלוק הנוכחי הוא הראשון
    # בשרשרת (נוצר הכי מוקדם), אז השדה מכיל בלוק ריק
    # - null. #
    input: Block
    # הבלוק הכולל את הקלט לעיסקה זו - העיסקה שבה נוצר
    # או נמסר המטבע שמשלמים בעיסקה זו. אם אין קלט כי המטבע
    # נוצר בעיסקה הנוכחית, אז שדה זה מכיל בלוק ריק
    # - null #
    receiver: str
    # המפתח הציבורי שאליו מועבר המטבע בעיסקה זו #
```

בנוסף, נתון המשתנה הגלובאלי latest המייצג את הבלוק האחרון בשרשרת (זה שנוצר הכי מאוחר).

א [10 נק']. כיתבו פונקציה המקבלת מפתח ציבורי ומחשבת את ה"יתרה" שלו – בכמה מטבעות לכל היותר הוא יכול להשתמש:

```
def balance(key: str)->int: ...
```

מותר להוסיף שדות למחלקה Block לפי הצורך.

ב [10 נק']. תארו שרשרת בלוקים שבה שלושה אנשים: לאחד יתרה 2, לשני יתרה 1, ולשלישי יתרה 0. הדגימו את פעולת הפונקציה balance על כל אחד מהשלושה.

פתרון: א. היתרה של מפתח ציבורי שווה לסכום הכסף שקיבל ועדיין לא ניצל. קודם-כל צריך לבדוק איזה בלוקים כבר נוצלו:

```
def calculate_used_blocks()->set:
    result = {}
    current = latest
```



```

while current is not null:
    result.add(current.input)
    current = current.previous
return result

```

עכשיו נבדוק מה היתרה:

```

def balance(key:str)→int:
    used_blocks = calculate_used_blocks()
    sum = 0
    current = latest
    while current is not null:
        if current.receiver!=key:
            break
        if current in used_blocks:
            break
        sum += 1
        current = current.previous

```

ב. נניח שהשרשרת נראית כך (מהמאוחר אל המוקדם):

```

latest == block4;
block4 == {
    previous: block3,
    input: block2,
    receiver: Ami
}
block3 == {
    previous: block2,
    input: null,
    receiver: Rami
}
block2 == {
    previous: block1,
    input: null,
    receiver: Tami
}
block1 == {
    previous: null,
    input: null,
    receiver: Ami
}

```

}

יש רק בלוק אחד שנוצל, והוא בלוק 2.

לכן, אם מריצים balance על עמי, רואים שהוא נמצא בשני בלוקים שלא נוצלו – 1 ו 4, ולכן היתרה שלו 2.

אם מריצים balance על תמי, רואים שהיא לא נמצאת באף בלוק שלא נוצל – ולכן היתרה שלה 0.

אם מריצים balance על רמי, רואים שהוא נמצא רק בבלוק אחד שלא נוצל – ולכן היתרה שלו 1.