

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

## שם הקורס: אלגוריתמים כלכליים

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.

## 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 הפירסומות 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	
100	200	400	900	אורך הפרסומת בשניות
3	5	7	9	הערך למפרסם באלפי ש"ח

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

### 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]
```

## **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.

## 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.

## שאלה 6. תוספת נקודות

אם שמכם הופיע באחד או יותר מקבצי הפתרונות שפורסמו בגיטהאב של הקורס, אנא ציינו את מספרי המטלות. כל מטלה שבה שמכם הופיע, מזכה אתכם בתוספת של נקודה לציון הסופי.