



פקולטה: מדעי הטבע
מחלקה: מדעי המחשב
שם הקורס: אלגוריתמים 2
קוד הקורס: 2-7027110 קבוצה 4
מועד ____ ב ____ סמסטר ____ ב ____ תאריך בחינה: 19/7/2018
משך הבחינה: 3 שעות
שם המרצה: אראל סגל-הלוי

פתרון

שאלה 1 [20 נק']. מציאת שיווי משקל

נתונה המחלקה הבאה, המייצגת שחקן במשחק כלשהו לשני שחקנים:

```
class Player {  
    float utility(int myAction, int otherAction);  
  
    // מחזירה את התועלת של השחקן מהתוצאה המתקבלת כאשר השחקן משחק בפעולה מס' myAction  
    // והשחקן השני משחק בפעולה מס' otherAction.  
};
```

א [10 נק']. כיתבו פונקציה המקבלת כקלט שני שחקנים וכמה פעולות אפשריות יש לכל שחקן, ומוצאת את כל שיווי-משקל-נאש של המשחק (באסטרטגיות טהורות בלבד - בלי הגרלות). מספרי הפעולות של שחקן 1 הן בין 0 לבין $\text{numActions1}-1$, ושל שחקן 2 בין 0 לבין $\text{numActions2}-1$.

פתרון אפשרי (לא בהכרח היעיל ביותר): קודם נכתוב פונקציה הבודקת האם זוג מסויים של פעולות הוא ש"מ נאש:

```
bool isPureNashEquilibrium(Player player1, Player player2,  
    int numActions1, int numActions2, int action1, int action2) {  
    float utility1 = player1.utility(action1, action2);  
    float utility2 = player2.utility(action2, action1);  
    // בדוק אם שחקן 1 יכול למצוא פעולה טובה יותר:  
    for (int i=0; i<numActions1; ++i) {  
        float otherUtility1 = player1.utility(i, action2);  
        if (otherUtility1 > utility1)  
            return false;  
    }  
    // בדוק אם שחקן 2 יכול למצוא פעולה טובה יותר:  
    for (int i=0; i<numActions2; ++i) {  
        float otherUtility2 = player2.utility(action1, i);
```

```

        if (otherUtility2 > utility2)
            return false;
    }
    // בנקודה זו אף שחקן לא יכול למצוא פעולה טובה יותר, לכן זה ש"מ
    return true;
}

```

עכשיו נשאר רק להריץ את הפונקציה הנ"ל על כל הזוגות האפשריים של פעולות:

```

void findPureNashEquilibria(Player player1, Player player2,
    int numActions1, int numActions2) {
    for (int action1=0; action1<numActions1; ++action1) {
        for (int action2=0; action2<numActions2; ++action2) {
            if (isNashEquilibrium(player1, player2, numActions1,
numActions2, action1, action2)) {
                println("Player 1 plays " + action1 + " and player
2 plays " + action2);
            }
        }
    }
}

```

ב [10 נק']. ציירו טבלת-משחק של משחק שיש בו בדיוק ארבעה שיווי-משקל טהורים, והדגימו עליה את הפונקציה מסעיף א.

פתרון: למשל משחק-תיאום - קבעתם פגישה בבניין בן 4 קומות ואתם לא זוכרים באיזו קומה. התועלות של שני השחקנים הן:

```

1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

```

האלגוריתם יעבור על כל השורות ועל כל הטורים, ויבדוק לגבי כל משבצת אם היא ש"מ נאש. הבדיקה תתבצע ע"י בדיקה אם יש מספר גבוה יותר בטור או בשורה. תשובה חיובית תתקבל בדיוק בארבעת המשבצות שבאלכסון הראשי.

שאלה 2 [30 נק']. בדיקת ביטחון עצמי של תלמידים

המורה לספרות רצתה למדוד את רמת הביטחון העצמי של התלמידים בכיתה. היא שאלה כל אחד מהתלמידים "מה לדעתך יהיה הציון שלך בבחינה הקרובה"? אבל היתה לה תחושה שהתשובות של התלמידים לא אמיתיות. לאחר מחשבה מרובה, היא החליטה להשתמש במנגנון הבא:

- כל תלמיד יכתוב על פתק, מה לדעתו יהיה הציון שלו בבחינה הקרובה.
- התלמיד שיכתוב את הציון הנמוך ביותר, לא ייבחן כלל; הציון שלו יהיה הציון הבא ברשימה (השני מלמטה).
- שאר התלמידים ייבחנו כרגיל.

א [10 נק']. הוכיחו שהמנגנון של המורה לספרות הוא אמיתי - כלומר הפעולה הטובה ביותר עבור כל תלמיד היא להגיד את הציון שהוא באמת חושב שיקבל בבחינה. יש לכתוב הוכחה מלאה ומפורטת. אין להשתמש במילים "כפי שלמדנו בהרצאות" - זכרו שמדובר בתלמידים לספרות, הם לא היו בהרצאות!

פתרון: נניח שאתה תלמיד וחושב שתקבל ציון x . נבדוק כמה אפשרויות.

- נניח שהציון שלך הכי נמוך, והציון השני מלמטה הוא y . אם תגיד את האמת - תקבל y . אם תגיד ציון כלשהו שהוא נמוך יותר מ- y - עדיין תהיה הכי נמוך בכיתה ותקבל ציון y . אם תגיד ציון גבוה יותר מ- y - כבר לא תהיה הכי נמוך - תצטרך להיבחן ותקבל ציון x שהוא נמוך מ- y . כך שאתה אף פעם לא מרויח, ועלול להפסיד.
- נניח שהציון שלך לא הכי נמוך - והציון הכי נמוך הוא y . אם תגיד את האמת - תיבחן ותקבל x . אם תגיד ציון כלשהו שהוא גבוה יותר מ- y - עדיין לא תהיה הכי נמוך בכיתה, תיבחן ותקבל x . אם תגיד ציון נמוך יותר מ- y - אתה כבר תהיה הכי נמוך, לא תצטרך להיבחן, אבל תקבל ציון y שהוא נמוך יותר מ- x . גם כאן אתה אף פעם לא מרויח, ועלול להפסיד.

ב [10 נק']. המורה ללשון ראתה את המנגנון של המורה לספרות והחליטה להשתמש במנגנון דומה, בהבדל אחד: התלמיד שיכתוב את הציון הגבוה ביותר, לא יצטרך להיבחן כלל; הציון שלו יהיה הציון הבא ברשימה (השני מלמעלה). שאר התלמידים ייבחנו כרגיל. האם המנגנון של המורה ללשון הוא אמיתי? כיתבו הוכחה מפורטת או הביאו דוגמה נגדית.

- **פתרון:** המנגנון אינו אמיתי. לדוגמה, נניח שיש שני סטודנטים, הראשון חושב שיקבל 80 והשני חושב שיקבל 70. אם השני יגיד את האמת, הוא יצטרך להיבחן ויקבל (לפי דעתו) 70. אבל אם יתחכם ויגיד 100, הוא יהיה הכי גבוה בכיתה, ויקבל אוטומטית 80.

ג [10 נק']. גם המורה להסטוריה ראה את המנגנון של המורה לספרות (המנגנון מסעיף א) והחליט להשתמש במנגנון דומה, אבל הוא רצה להשיג מטרה נוספת - לבדוק כמה שפחות בחינות. הניחו שבכיתה יש 30 תלמידים. תארו מנגנון אמיתי, שיעודד כל תלמיד להגיד את הציון שהוא באמת חושב שיקבל בבחינה, ויחד עם זה, ייתן פטור מהבחינה לכמה שיותר תלמידים. הסבירו בפירוט כמה ואיזה תלמידים ייבחנו, כמה ואיזה תלמידים יקבלו פטור מהבחינה, ואיזה ציון כל תלמיד יקבל. הוכיחו שהמנגנון אמיתי.

פתרון: תלמיד אחד ייבחן - זה שאמר את הציון הגבוה ביותר בכיתה (אם יש שניים או יותר - נבחר אחד מהם בהגרלה). שאר 29 התלמידים יקבלו פטור מהבחינה, ויקבלו אוטומטית את הציון הגבוה ביותר. הוכחה שהמנגנון אמיתי:

- נניח שהציון שלך x לא הכי גבוה, והציון הכי גבוה הוא y . אם תגיד את האמת - תהיה פטור מהמבחן ותקבל y . אם תגיד ציון כלשהו שהוא נמוך יותר מ- y - עדיין תקבל y . אם תגיד ציון גבוה יותר מ- y - תהיה הכי גבוה - תצטרך להיבחן ותקבל x שהוא נמוך מ- y . כך שאתה אף פעם לא מרויח, ועלול להפסיד.
- נניח שהציון שלך x הכי גבוה, והציון השני מלמעלה הוא y . אם תגיד את האמת - תיבחן ותקבל x . אם תגיד ציון כלשהו שהוא גבוה יותר מ- y - עדיין תהיה הכי גבוה בכיתה, תיבחן ותקבל x . אם תגיד ציון נמוך יותר מ- y - אתה כבר לא תהיה הכי גבוה, לא תצטרך להיבחן, אבל תקבל ציון y שהוא נמוך יותר מ- x . גם כאן אתה אף פעם לא מרויח, ועלול להפסיד.

שאלה 3 [20 נק']. חלוקה הוגנת של משאבי חישוב

עמי תמי וצומי רוצים להשתמש במחשב-העל המחלקתי לצורך ביצוע חישובים מורכבים. הערך של עמי הוא: $1 * \text{כמות}$ הדיסק שהוא מקבל ועוד $2 * \text{כמות המעבד שהוא מקבל ועוד } 3 * \text{כמות הזיכרון שהוא מקבל}$. הערכים של תמי ושל צומי נקבעים באופן דומה רק עם מספרים שונים, בהתאם לטבלה הבאה (כאשר t הוא פרמטר כלשהו):

דיסק	מעבד	זיכרון	
1	2	3	עמי
4	5	6	תמי
t	8	7	צומי

כיתבו תוכנית, בשפת-תיכנות כלשהי לבחירתכם, המוצאת חלוקה של משאבי המיחשוב כפונקציה של t . החלוקה צריכה לקיים את התנאים הבאים:

- יעילות פארטו.
- אין קנאה - אף אחד מהסטודנטים לא מקנא במשאבים שקיבלו סטודנטים אחרים.

פתרון: האלגוריתם המתאים כאן הוא מיקסום מכפלת הערכים. כדי לבצע אותו ניתן להשתמש בפקודה הבאה בשפת

Mathematica [כמו במטלה 8 פרט לשימוש בפרמטר t במקום 9]:

```
FindMaximum[{Log[3 m1 + 2 p1 + d1] + Log[6 m2 + 5 p2 + 4 d2] +
Log[t (1 - m1 - m2) + 8 (1 - p1 - p2) + 7 (1 - d1 - d2)]},
0 <= m1 + m2 <= 1, 0 <= d1 + d2 <= 1, 0 <= p1 + p2 <= 1,
0 <= m1 <= 1, 0 <= m2 <= 1, 0 <= p1 <= 1, 0 <= p2 <= 1,
0 <= d1 <= 1, 0 <= d2 <= 1}, {{m1, 0.1}, {m2, 0.1}, {p1, 0.1}, {p2, 0.1}, {d1, 0.1}, {d2, 0.1}}]
```

שאלה 4 [30 נק'] . חלוקה הוגנת של ירושה עם בתים

איל הנדל"ן הנודע דונאלד דאק נפטר בשיבה טובה והוריש לארבעה ילדיו ארבעה בתים וכן סכום-כסף נכבד.

עליכם לעזור להם לחלק ביניהם את הירושה בהתאם לדרישות הבאות:

- כל אחד מהיורשים מקבל בית אחד בדיוק (וסכום-כסף כלשהו).
- אף יורש לא מקנא ביורשים האחרים.
- כל העיזבון מחולק בין היורשים.

א [10 נק'] . כיתבו אלגוריתם, בעברית או בפסאודו-קוד, הפותר את הבעיה. מותר להשתמש באלגוריתמים שנלמדו בכיתה, אולם יש להוכיח שהם אכן פותרים את הבעיה.

פתרון: נשתמש באלגוריתם חלוקת שכר דירה - המודל הקרדינלי (שבוע 8):

- נשתמש באלגוריתם ההונגרי כדי למצוא את ההשמה הממקסמת את סכום הערכים.
- נשתמש בתוצאות-הביניים של האלגוריתם ההונגרי כדי לחשב, לכל אחד מהיורשים, כמה הוא צריך לשלם/לקבל כך שההשמה תהיה ללא קנאה (כפי שנלמד בכיתה);
- בסוף, נחלק את הכסף שנשאר שווה בשווה בין כל היורשים. כיוון שהחלוקה הקודמת היתה ללא קנאה, היא תישאר ללא קנאה גם אחרי שנוסיף לכל שחקן סכום-כסף קבוע.

ב [20 נק'] . הדגימו את האלגוריתם שכתבתם בסעיף א על מצב שבו ערכי הבתים בעיני היורשים הם (לפי הסדר):

- בית: א ב ג ד
- מיקי: 80, 60, 40, 20
- דייזי: 90, 70, 50, 30
- גופי: 60, 80, 20, 40
- פלוטו: 70, 90, 30, 50

וסכום הירושה הוא 1000. תארו בפירוט את פעולת האלגוריתם. הסבירו מי מקבל כל בית ואיך מתחלק הכסף.

פתרון: הרצה של האלגוריתם ההונגרי ניתן לראות כאן:

<http://www.hungarianalgorithm.com/solve.php?c=80-60-40-20--90-70-50-30--60-80-20-40--70-90-30-50&obj=max>

ההשמה המתקבלת היא:

- מיקי - ד
- דייזי - ב
- גופי - ג
- פלוטו - א

סכום הערכים הוא 260.

(אפשר גם להחליף בין מיקי לדייזי ו/או בין גופי לפלוטו - יתקבל אותו סכום).

תוך כדי ההרצה, מורידים 40 מעמודות א, ב. לכן יש לשלם 40 לדייזי ולפלוטו והחלוקה הופכת להיות ללא-קנאה.

הירושה שנשארה היא 920 - יש לחלק לכל שחקן רבע מזה, כלומר 230.

בסה"כ, החלוקה היא:

- מיקי - בית ד ו-230 ש"ח.
- דייזי - בית ב ו-270 ש"ח.
- גופי - בית ג ו-230 ש"ח.
- פלוטו - בית א ו-270 ש"ח.

שאלה 5 [20 נק'] חוזה חכם לפתרון בעיית החלוקה

בבעיית חלוקת המספרים, נתון מערך של מספרים שלמים. צריך לחלק אותו לשני תת-מערכים, כך שסכום המספרים בשני המערכים הוא זהה (בהנחה שאכן קיימת חלוקה כזאת).

למשל: אם הקלט הוא המערך $\{1,2,3,4\}$, אז פלט אפשרי הוא שני המערכים: $\{1,4\}$ ו $\{2,3\}$.

כיוון שהבעיה היא קשה חישובית, אנחנו רוצים לכתוב חוזה חכם במערכת אתריום, שיעודד אנשים לפתור את הבעיה בעצמם ויתן פרס של 1 אֶתֶר לפותר הראשון.

לפניכם שלד של חוזה בשפת solidity (להזכרכם, `int[]` הוא מערך של מספרים שלמים, כמו ב-Java):

```
contract Partition {
    constructor(int[] input) public {
        ....
    }

    function solve(int[] part1, int[] part2) public {
        ....
    }
}
```

- הבנאי של החוזה מקבל כקלט את המערך שיש לחלק.
 - הפונקציה `solve` היא הפונקציה שהפותרים צריכים לקרוא לה כדי להציע את הפתרון שלהם לבדיקה. השלימו את שתי הפונקציות החסרות. הוסיפו משתנים ופונקציות נוספים לפי הצורך.
- אם אינכם זוכרים את התחביר של שפת solidity, הניחו שהתחביר זהה לשפת Java או C++ לפי בחירתכם. כמו כן, אתם יכולים להשתמש בכל הפונקציות הנמצאות בספריות התקניות של השפות הללו. הסבירו היטב בעברית מה אתם עושים.

פתרון: הרעיון המרכזי של חוזה חכם הוא, שכל אחד יכול לפנות אליו ולשלוח אליו פתרונות, אבל החוזה בודק ומוודא שהפתרון הוא נכון, כך שרק מי שפתר נכון יקבל את הפרס. לשם כך, הבנאי צריך לזכור את מערך הקלט, והפונקציה `solve` צריכה לבדוק שני דברים: (א) הפתרון שנשלח הוא אכן חלוקה של מערך הקלט, (ב) הפתרון שנשלח אכן כולל שתי קבוצות עם סכום שווה. בנוסף צריך לוודא שרק הפותר הראשון מקבל את הפרס; אפשר לעשות זאת ע"י דגל בוליאני, או פשוט להרוס את החוזה בסוף הפונקציה `solve`.

הנה פתרון אפשרי. חלק מהפונקציות הושמטו כי המימוש שלהן אלמנטרי (גם במבחן לא היה צורך לממש אותן - היה אפשר להניח שהן ממומשות בספריה כלשהי של השפה):

```
contract Partition {
    int[] public input;

    constructor(int[] newInput) public {
        input = newInput;
    }
}
```



```

        sort(input);
    }

    // calculate the sum of the given array of ints:
    function sum(int[] array) returns(int) { ... }

    // sort the given array of ints:
    function sort(int[] storage arr) { ... }

    // push all elements in array2 to the end of array1:
    function addAll(int[] storage array1, int[] array2) { ... }

    function solve(int[] part1, int[] part2) public {
        // verify that the sum of the two parts is the same:
        require(sum(part1)==sum(part2));

        // verify that the parts indeed partition the input:
        int[] parts12;
        addAll(parts12, part1);
        addAll(parts12, part2);
        sort(parts12);

        require(parts12.length==input.length);
        for (uint i=0; i<parts12.length; ++i)
            require(parts12[i]==input[i]);

        // It is correct - send the prize to the happy solver:
        msg.sender.transfer(1 ether);
    }

```

```
    // Destroy the contract,  
    // so that people will not try to solve it again.  
    selfdestruct();  
}  
}
```