

מטלה - חלוקת עלויות ומכרזי שירות

יש לענות על שאלה אחת לבחירתכם.

שאלה 1: חלוקת עלויות לבניית מעלית

בבניין עם n קומות, הדיירים החליטו לבנות מעלית.

העלות של בניית מעלית בגובה k קומות היא k אלף ש"ח.

הדיירים מעוניינים לחלק את העלות ביניהם בעזרת ערך שאפלי.

כמה תשלם כל קומה?

הדרכה: חשבו קודם את הערך עבור $n=2$ ו $n=3$, ואז הכלילו ל- n כלשהו.

- פתרון: ירדן בן אמיתי (המקרים הפרטיים) ושהם סעדיה (המקרה הכללי).

שאלה 2: חלוקת עלות נסיעה לאוניברסיטה

שאלה זו מיועדת לצוותים עם 3 חברים ומעלה (אפשר להתחבר עם צוות נוסף לצורך השאלה).

א. לכל אחד מחברי הצוות, חשבו בקירוב את עלות הנסיעה במונית מהבית לאוניברסיטה. לצורך החישוב

אפשר להשתמש ב-Google Maps. אם אתם לא מצליחים לחשב עלות מדוייקת, חשבו את זמן

הנסיעה לפי Waze והכפילו בעלות נסיעה במונית לפי מונה.

ב. חשבו את עלות הנסיעה כאשר כל חברי הצוות נוסעים יחד, במסלול הקצר ביותר העובר דרך כולם.

ג. חשבו כמה כל אחד יצטרך לשלם לפי ערך שאפלי. האם זה משתלם?

- פתרון: יערה אתי ושי.

שאלה 3: איחוד מפלגות

[נכתב לקראת הבחירות לכנסת, אדר א ה'תשע"ט]

שלוש מפלגות קטנות שוקלות להתאחד למפלגה אחת לקראת הבחירות, אבל הן לא מצליחות להסכים על

קביעת המקומות ברשימה המשותפת. סקרי דעת-קהל אמינים מראים ש:

- כל אחת מהמפלגות לא עוברת את אחוז החסימה כשהיא רצה לבד.
 - המפלגה המאוחדת מקבלת 10 מנדטים.
 - אם רק מפלגות א+ב מתאחדות, הן מקבלות 4 מנדטים; ב+ג - 5 מנדטים, ג+א - 6 מנדטים.
- עיזרו למפלגות למצוא פתרון הוגן לחלוקת המקומות בעשירה הראשונה.

- פתרון: דניאל ואיתן.

שאלה 4: פונקציה תת-מודולרית

תזכורת: פונקציה v על קבוצות נקראת תת-מודולרית (submodular) אם, לכל שתי קבוצות S, T ולכל

איבר i , יש עלות שולית פוחתת:

If $S \leq T$, then $v(S \cup \{i\}) - v(S) \geq v(T \cup \{i\}) - v(T)$

הגדרה: פונקציה v על קבוצות נקראת תת-חיצורית (subadditive) אם, לכל שתי קבוצות S, T :

$$v(S \cup T) \leq v(S) + v(T)$$

א. תנו דוגמה לפונקציה שהיא גם תת-אדיטיבית וגם תת-מודולרית.

ב. תנו דוגמה לפונקציה שהיא תת-אדיטיבית אבל לא תת-מודולרית.

ג. תנו דוגמה לפונקציה שהיא לא תת-אדיטיבית ולא תת-מודולרית.

ד. הוכיחו שכל פונקציה תת-מודולרית היא תת-אדיטיבית.

אם אתם מסתבכים תסתכלו כאן:

https://en.wikipedia.org/wiki/Utility_functions_on_indivisible_goods

• פתרון:

שאלה 5: מכרז מולין-שנקר - דוגמאות

א. הראו דוגמה עם 2 או 3 שחקנים, שבה כלל-התשלום של שאפלי אינו מונוטוני. הראו את טבלת העלויות, חשבו את ערכי שאפלי לכל תת-קבוצה, והסבירו מדוע זה לא מונוטוני.

ב. הראו דוגמה עם 2 או 3 שחקנים, שבה מכרז מולין-שנקר עם כלל-התשלום של שאפלי אינו אמיתי (אפשר להיעזר בסעיף א). הסבירו בפירוט מה יעשה המכרז בדוגמה זו, ומדוע זה לא אמיתי.

ג. הראו דוגמה עם 2 או 3 שחקנים, שבה מכרז מולין-שנקר עם כלל-התשלום של שאפלי אינו יעיל-פארטו. הסבירו בפירוט מה יעשה המכרז בדוגמה זו, ומדוע זה לא יעיל-פארטו.

ד [בונוס – למי שעדיין לא זכה הסמסטר]. היכנסו לויקיפדיה האנגלית, לדף "Cost-sharing mechanism". מיצאו את תת-הכותרת "Binary service, decreasing marginal costs". הוסיפו את הדוגמאות שלכם במקומות המתאימים בדף (אם אתם לא בטוחים איפה בדיוק להוסיף, תשאלו אותי).

הצוות הראשון שיבצע משימה זו, יקבל 2 נקודות תוספת לציון הבחינה. כדי לזכות בנקודות, אחד מחברי הצוות צריך לפתוח חשבון ויקיפדיה בשמו המלא, כך שהעריכות יירשמו על שמו בהסטוריית העריכה.

עצה: כדי שהעריכות שלכם יישארו בדף, ולא יימחקו ע"י עורכים אחרים, יש להסביר באופן ברור ובאנגלית תקנית. לפני שאתם מתחילים אנא ודאו שלא עשו זאת לפניכם; הבונוס יינתן רק לראשון.

• פתרון:

שאלה 6: מכרז SN

מכרז SN בוחר תת-קבוצה של נוסעים באופן הבא:

• הרץ VCG : בחר את תת-הקבוצה הממקסמת את סכום ערכי-הנוסעים פחות עלות הנהג, וחשב את התשלום p_i שכל שחקן i צריך לשלם לפי נוסחת VCG .

• חשב את סכום התשלומים של כל השחקנים; נסמן סכום זה ב S .

ברוך ה' חנוך הדעת

- חשב את העלות של תת-הקבוצה הנבחרת; נסמן עלות זו ב C .
 - גבה מכל שחקן i תשלום $p_i * C/S$.
- מטרת ההכפלה בשלב האחרון היא לוודא שהמכרז הוא מאוזן תקציבית – סכום התשלומים של כל השחקנים אחרי ההכפלה שווה לעלות.
- א. האם מכרז SN אמיתי? הוכיחו או הפריכו.
- ב. האם מכרז SN יעיל פארטו (בהנחה שכל השחקנים אומרים אמת)? הוכיחו או הפריכו.
- **פתרון:**

שאלה 7: תיכנות - ערך שאפלי

- א. מצאו בשפת python (או בשפה אחרת החביבה עליכם) פונקציית-ספרייה המאפשרת לעבור על כל הסדרים האפשריים (פרמוטציות) של קבוצה נתונה.
- ב. השתמשו בפונקצייה זו על-מנת לכתוב פונקציה המחשבת את ערך שאפלי.
- **פתרון:** ראו קוד בעמוד הבא

```

#include <algorithm>
#include <functional>
#include <numeric>
#include <map>
#include <set>
using namespace std;

map<int,double> shapley_values(int num_of_players,
                             map<set<int>,double> map_subset_to_cost) {
    vector<int> players(num_of_players);
    iota(players.begin(), players.end(), 1); // fill with 1,...,n
    map<int, double> map_player_to_sum_of_marginal_costs;
    int num_permutations = 0;
    do {
        // calculate marginal costs for a specific permutation:
        double current_cost = 0;
        set<int> current_subset;
        for (int player: players) { // loop over current permutation
            current_subset.insert(player);
            double new_cost = map_subset_to_cost[current_subset];
            double marginal_cost = new_cost - current_cost;
            map_player_to_sum_of_marginal_costs[player] += marginal_cost;
            current_cost = new_cost;
        }
        ++num_permutations;
    } while ( next_permutation(players.begin(), players.end()) );
    for (auto& [player,cost]: map_player_to_sum_of_marginal_costs)
        cost /= num_permutations;
    return map_player_to_sum_of_marginal_costs;
}

int main() {
    map<set<int>,double> map_subset_to_cost = {
        { {}, 0 },
        { {1}, 1000 },
        { {2}, 2000 },
        { {1,2}, 2000 },
        { {3}, 3000 },
        { {1,3}, 3000 },
        { {2,3}, 3000 },
        { {1,2,3}, 3000 },
    };
    auto sv = shapley_values(3, map_subset_to_cost);
    cout << sv << endl;
}

```