אלגוריתמים כלכליים

מבוסס על הרצאות ,סיכומים ,מצגות של ד"ר סגל הלוי דוד אראל:
https://github.com/erelsgl/ariel-algorithms-5779

23/7/2019

שי נאור

חלוקה הוגנת (שיעור 1):

קלט: משאב משותף, זכויות שוות.
מטרה: חלוקה שתהיה הוגנת בעיני כולם.

חלוקת עוגה בין שני ילדים - חתוך ובחר:

האלגוריתם:
1. אחד הילדים מחלק את העוגה לשני חלקים שווים בעיניו.
2. הילד השני בוחר את החלק הטוב בעיניו.
3. הילד שחילק מקבל את החלק שנותר.

תכונות:
1. כל משתתף חושב שהחלוקה שלו שווה לפחות ½ – חלוקה פרופורציונלית (proportional).
2. כל משתתף חושב שהחלק שלו טוב לפחות כמו כל שאר החלקים – חלוקה ללא קנאה(envy-free).

חלוקת עוגה מודל כלכלי:
העוגה C היא קטע(חד-מימדי) או מצולע (רב-מימדי).
לכל משתתף יש פונקציית צפיפות ערך על העוגה:
$Vi: C \rightarrow R$
ערך של חתיכת עוגה הוא אינטגרל על צפיפות הערך:
$V_i(X) = \int_x Vi(x)\, dx$

חלוקת עוגה להרבה אנשים:
תכונות:
1. כל אחד חושב שהחלק שלו שווה לפחות אחד חלקי n – חלוקה פורפורציונלית:
$V_i(X_i) >= V_i(C)/n$
2. כל אחד חושב שהחלק שלו טוב לפחות כמו כל האחרים – חלוקה ללא קנאה:
$V_i(X_i) >= V_i(X_j)$

שאלה: איזה מתי התכונות קשה יותר?
תשובה: חלוקה ללא קנאה היא קשה לפחות כמו חלוקה פרופורציונלית.
הוכחה: אם חלוקה היא ללא קנאה, אז כל אחד חושב שהחלק שלו טוב לפחות כמו כל שאר n החלקים. לפי כלל שובך היונים, כל אחד חושב שהחלק שלו שווה לפחות אחד חלקי n מהשווי הכללי. כלומר חלוקה ללא קנאה היא בהכרח גם חלוקה פרופורציונלית.

חלוקה פרופורציונלית של קרקע – אלגוריתם "המפחית האחרון":
האלגוריתם (רקורסיבי):
אם נשאר שחקן אחד – הוא מקבל את כל העוגה.
אם יש יותר משחקן אחד – מסדרים את השחקנים בסדר שרירותי, ואז:
- מבקשים מהראשון לסמן שטח השווה בעיניו בדיוק אחד חלקי n.
- עוברים על כל השחקנים, כל שחקן נשאל בתורו האם החלק שסומן שווה בעיניו לכל היותר אחד חלקי n?
  - אם הוא אומר כן ממשיכים הלאה.

- אחרת, מבקשים ממנו להפחית את החלק המסומן כך שישאר חלק קטן יותר השווה בעיניו בדיוק אחד חלקי n.
- בסופו של דבר השחקן האחרון שהפחית מהשטח, מקבל את השטח המסומן והולך הביתה.
- שאר(n−1) השחקנים ממשיכים לחלק את העוגה ביניהם באופן רקורסיבי.

משפט: אלגוריתם "המפחית האחרון" נותן חלוקה פרופורציונלית. כלומר כל שחקן המשחק לפני הכללים מקבל לפחות 1 חלקי n מערך העוגה בעיניו.

סיבוכיות: $O(n^2)$.

חלוקה פרופורציונלית של קרקע – אלגוריתם "אבן – פז":

האלגוריתם(רקורסיבי):
- אם נשאר שחקן אחד הוא מקבל את כל העוגה.
- אם יש יותר משחקן אחד, ומספר השחקנים זוגי, אז:
  - כל שחקן מחלק לשני חלקים בשווי ½ בעיניו.
  - חותכים את העוגה בחציון של הקווים.
  - שולחים כל שחקן לחצי המכיל את הקו שלו.
  - מחלקים כל חצי באופן רקורסיבי.
- אם יש יותר משחקן אחד ומספר השחקנים הוא איזוגי, אז:
  - כל שחקן מחלק לשני חלקים ביחס של: $(n-1)/2$ ל- $(n+1)/2$
  - שולחים כל שחקן לחצי המכיל את הקו שלו.
  - מחלקים כל חצי באופן רקורסיבי.

משפט: אלגוריתם "אבן-פז" נותן חלוקה פרופורציונלית. כלומר כל שחקן המשחק לפני הכללים מקבל לפחות 1 חלקי n מערך העוגה בעיניו.

סיבוכיות: $O(n\log n)$.

שאלה: האם ניתן לשפר יותר את זמן הריצה?
תשובה: לא! הוכח בשנת 2007.

<u>חלוקה ללא קנאה:</u>
כפי שנאמר קודם לכן, חלוקה ללא קנאה היא קשה יותר מחלוקה פרופורציונלית.

חלוקה ללא קנאה ל-3 שותפים אלגוריתם "Selfridge – Conway":
האלגוריתם:
- שחקן א' חותך שלוש חתיכות השוות בעיניו.
- אם שחקנים ב' ו- ג' מעדיפים חתיכות שונות – סיימנו.
- אחרת, שחקן ג' מקצץ את החתיכה השווה ביותר בעיניו ומשווה לשניה בעיניו.
- השחקנים בוחרים חתיכה לפי הסדר: ב', ג', א'. שחקן ג' חייב לבחור בחתיכה שקיצץ אם לא נבחרה קודם ע"י שחקן ב'.
- קיבלנו חלוקה ללא קנאה עם שארית.
- שחקן ב' או ג' בחרו את החתיכה המקוצצת.
- מי מביניהם שלא בחר את החתיכה המקוצצץ מחלק את השארית לשלוש חתיכות השוות בעיניו.

– השחקנים בחורים חתיכה מהשארית באופן הבא: השחקן שלא חילק את השארית(ב' או ג'), שחקן א', השחקן שחילק את השארית.

משפט: אלגוריתם "Selfridge – Conway" נותן חלוקה ללא קנאה. כלומר, כל שחקן המשחק לפני הכללים מקבל חתיכה טובה לפחות כמו שתי האחרות בעיניו.

שאלות:
1. מה קורה כשיש 4 שותפים או יותר?
2. איך מוצאים חלוקה ללא קנאה עם חתיכות קשירות?

תשובות:
1. עבור 4 שחקנים ישנם מספר אלגוריתמים לחלוקה ללא קנאה, אבל הם מסובכים מאוד.
2. עבור פרוסות קשירות, לא קיים אלגוריתם סופי המוצא חלוקה ללא קנאה, אפילו עבור שלושה שחקנים.

חלוקה כמעט ללא קנאה, קשירה:
הרעיון הוא להסתכל על כל החלוקות הקשירות ל-N חתיכות.
כל חלוקה מוגדרת על ידי N מספרים שסכומם הוא קבוע.
מרחב החלוקות הקשירות הוא:
עבור N= 2 – קטע.
עבור N= 3 – משולש.
עבור N= 4 – טטראדר.
באופן כללי – סימפלקס.

בכל נקודה בסימפלקס החלוקות, ניתן לשאול כל שחקן "איזו חתיכה אתה הכי רוצה?".
התשובה לכך היא מספר בין 1 ל N.
חלוקה ללא קנאה היא, נקודה שבה כל שחקן כותב מספר אחר.
חלוקה כמעט ללא קנאה היא, סימפלקסון שבו אפשר לחלק קודקוד לכל שחקן, כך שכל שחקן כתב על הקודקוד שלו מספר אחד.

אלגוריתם "סימונס 1999 ,Su":
– מחלקים את סימפלקס החלוקות לסימפלקסונים.
– נותנים כל צומת לשחקן, כולם מיוצגים.
– כל שחקן כותב, בכל צומת שלו, את מספר החתיכה הכי טובה בעיניו.
– מחפשים סיפלקסון עם N מספרים שונים.
– סימפלקסון כזה מהווה חלוקה קשירה כמעט ללא קנאה.

<table>
<tr><th>שחקנים</th><th>חלוקה פרופורציונלית</th><th>חלוקה ללא קנאה</th><th>חלוקה קשירה ללא קנאה</th></tr>
<tr><td>2</td><td colspan="3">2 שאילתות</td></tr>
<tr><td>3</td><td rowspan="3">$\Theta(n \log n)$</td><td>5</td><td rowspan="3">אינסוף!</td></tr>
<tr><td>4</td><td>200</td></tr>
<tr><td>n</td><td>$\Omega(n^2)$
$O(n^{nnnnn})$</td></tr>
</table>

<u>מטלה – חלוקה הוגנת של קרקעות ועוגות</u>

שאלה 1: **חלוקה פרופורציונלית בשני מימדים**

א. נתונה חלקת-אדמה בצורת מלבן. תארו אלגוריתם הנותן חלוקה פרופורציונלית של העוגה בין n אנשים, כך שכל אחד מקבל מלבן.

ב. הסבירו מהי צורה *קמורה* ותנו דוגמאות (ראו בויקיפדיה: https://en.wikipedia.org/wiki/Convex_set)

ג. נתונה חלקת-אדמה דו-ממדית קמורה. תארו אלגוריתם הנותן חלוקה פרופורציונלית של העוגה בין n אנשים, כך שכל אחד מקבל פרוסה קמורה.

תשובה 1:

א. נשתמש באלגוריתם "המפחית האחרון" עם ההגבלה הבאה: ניתן לבצע חיתך של המלבן אך ורק בצורה אנכית. אלגוריתם "המפחית האחרון" מקנה לנו החלוקה הפרופוריונלית, הדרישה על אופן החיתוך של המלבן מקנה לנו חלוקה שבא כל אחד יקבל מבלבן.

ב. במתמטיקה, קבוצת נקודות במרחב וקטורי היא קמורה אם לכל שתי נקודות שבתוכה, גם הקטע המחבר את שתי הנקודות נמצא כולו בתוכה. למשל, משולש, עיגול או מקבילית הן צורות קמורות, אבל טבעת או פרסה אינן צורות קמורות. (מתוך ויקיפדיה).

ג. נשתמש באלגוריתם "המפחית האחרון" עם ההגבלה הבאה: ניתן לבצע חיתך של הצורה הקמורה אך ורק על ידי קו ישר, וכך הצורות שהתקבלו על ידי החיתוך יהיו גם הן קמורות. אלגוריתם "המפחית האחרון" מקנה לנו החלוקה הפרופוריונלית, הדרישה על אופן החיתוך של הצורה הקמורה מבטיחה לנו שכל הצורות שהתקבלו יהיו קמורות גם.

שאלה 2: **חלוקת תורנויות פרופורציונלית**

אמא ואבא נסעו לנופש, והשאירו את n ילדיהם הגדולים לשמור על התינוקת ה-$n+1$. הילדים לא כל כך רוצים לשמור אבל אין להם ברירה, אז הם החליטו לחלק ביניהם את הזמן ולעשות תורנויות. לכל ילד יש העדפות שונות לגבי הזמן ביום שהכי קשה לשמור בו - יש כאלה שהכי קשה להם לשמור בלילה, יש כאלה שהכי קשה להם לשמור דווקא בצהריים, וכו'.

א. הגדירו בפירוט מהי חלוקה פרופורציונלית במצב זה (מדוע זה שונה ממה שלמדנו בכיתה?).

ב. כיתבו אלגוריתם המוצא חלוקה פרופורציונלית של זמן השמירה, כך שכל ילד מקבל תורנות רצופה.

תשובה 2:

א. חלוקה פרופורציונלית במקרה זה היא חלוקה שבה כל ילד יקבל חלק השווה לכל היותר אחד חלקי n בעניו מכל התורנויות. בכיתה למדנו חלוקה של דברים שאנחנו רוצים לקבל מהם כמה שיותר לכן הדרישה לפרופורציונליות הייתה אחד חלקי n לכל הפחות. מכיוון שנתון שהילדים לא כל כך רוצים לשמור על התינוקת (מחלקים משהו "שלילי") אז הדרישה לפרופורציונליות תהיה שכל ילד יקבל חלק השווה לכל היותר אחד חלקי n מכל התורנויות.

ב. נשתמש באלגוריתם "המפחית האחרון" עם כמה שינויים קלים:

- נסדר את הילדים באופן שרירותי כלשהו.
- נבקש מהילד הראשון לבחור קטע זמן (מההתחלה) השווה אחד חלקי n בעניו.

– נעבור על כל הילדים ונשאל אותם: "האם החלק שהוא בחר שווה בעיניהם לכל הפחות אחד חלקי n ?"

– אם כן ממשיכים לילד הבא.

– אחרת, מבקשים ממנו להוסיף זמן כך שהחלק שנבחר יהיה שווה לאחד חלקי n בעניו.

– האחרון שהוסיף זמן, התורנות שלו.

– ממשיכים את האלגוריתם עם n−1 ילדים באופן רקורסיבי.

שאלה 3: חלוקת תורנויות ללא קנאה

אמא ואבא נסעו לנופש וכו', כמו בשאלה 2.

א. תארו אלגוריתם שבו כל אחד מהילדים מקבל תורנות רצופה, והחלוקה היא ללא–קנאה–בקירוב (עד כדי שניה אחת).

ב. תנו דוגמה הממחישה את פעולת האלגוריתם של סעיף א.

תשובה 3:

א. נשתמש באלגוריתם של "סו" של חלוקת הסימפלקס לסימפלקסונים. המקנה חלוקה קשירה וללא קנאה בקירוב. הסבר: מכיוון שזמן התורנות הכולל הוא קבוע, אוסף כל החלוקות האפשריות הוא סימפלקס. כל נקודה בסימפלקס מייצגת חלוקה. עבור כל נקודה בסימפלקס אנחנו יכולים לשאול את כל אחד מהילדים איזה חלק הוא מעדיף. מכיוון שהחלוקה היא ללא קנאה בקירוב נחלק את הסימפלקס שלנו לסימפלקסונים קטנים, שאורך כל צלע הוא שניה. נשייך כל קודקוד לאחד הילדים, כך שבכל סימפלקסון כל ילד מיוצג. נעבור על כל קודקוד נשאל את הילד שהקודקוד שייך לו, איזה פרוסה הוא מעדיף ונסמן את המספר על הקודקוד. סימפלקסון שבו כל ילד אמר העדפה שונה היא חלוקה ללא קנאה בקירוב של שניה. ולפי הלמה של שפרנר בהכרח קיימת לפחות חלוקה אחת כזאת.

ב. נניח N = 3

השעות:	בוקר	צהריים	ערב
דני	10	0	0
רמי	0	10	0
דינה	0	0	10

האלגוריתם יחזיר את החלוקה שבה דני שומר בבוקר, רמי בצהריים, ודינה בערב.

שאלה 4: חלוקה עם זכויות לא שוות (unequal entitlements)

עמי ותמי עזרו לאמא להכין עוגה, אבל תמי עזרה יותר. עמי השקיע שעתיים ותמי השקיעה חמש שעות. אמא רוצה לחלק את העוגה ביניהם בצורה הוגנת בהתאם להשקעה.

א. תנו הגדרה הגיונית למושג "חלוקה פרופורציונלית" במצב זה.

ב. כיתבו אלגוריתם המוצא חלוקה פרופורציונלית.

תשובה 4:

א. השאלה מתייחסת לחלוקה פרופונציונלית בין שני ילדים עם זכויות לא שוות. ולכן כל אחד צריך לקבל חתיכה פרופורציונלית לזמן ההשקעה שלו. כלומר, ביחד הם השקיעו 7 שעות. תמי צריכה לקבל חתיכה בשווי 5/7 בעיניה מכל העוגה. ועמי צריך לקבל חתיכה בשווי 2/7 בעיניו מכל העוגה.

ב. האלגוריתם דומה מאוד לאלגוריתם "חתוך ובחר":
– נבקש מאחד מהם לבצע את חיתוך העוגה, נניח עמי, נבקש ממנו לחתוך את העוגה ל 7 חלקים השווים בעיניו.
– תמי, תבחר את 5 החלקים הטובים בעיניה.
– עמי יקח את שני החלקים הנותרים.
זוהי חלוקה פרופורציונלית.

שאלה 5: **שינוי חלוקה קיימת**
עמי רמי ותמי חילקו ביניהם עוגה בצורה פרופורציונלית (כל אחד קיבל לפחות 1/3). הם התיישבו לאכול, אבל לפני שהספיקו – נכנסה צומי וטענה שגם לה מגיע חלק. פתחו אלגוריתם המוצא חלוקה פרופורציונלית (כל אחד מקבל לפחות 1/4), ובנוסף, נותן לכל אחד משלושת הילדים הראשונים (עמי רמי ותמי) לפחות 3/4 מהערך שהיה לו בחלוקה הראשונה.

תשובה 5:
עמי, תמי ורמי כל אחד מהם יחלק את החתיכת עוגה שקיבל ל-4 חלקים השווים בעיניו.
צומי תבחר את החלקים הכי טובים בעיניה מבין 4 החלקים, של עמי, צומי ורמי, מכל אחד מהם.
מכיוון שעמי תמי ורמי כל אחד מהם חילק בעצמו את הפרוסה שלו ל-4 חלקים לא משנה איזה חלק צומי תקח מהם החלק שישאר להם שווה בעיניהם לפחות ¾ מהערך שהיה בחלוקה הראשונה. מכיוון שצומי תבחר מעמי, תמי ורמי את החלק הכי שווה בעניה מבין 4 החלקים שהם חתכו היא תקבל לפחות רבע מהשווי הכללי של העוגה בעניה.

שאלה 6: **חלוקה ללא קנאה עם שארית**
נתון האלגוריתם הבא לחלוקה בין 3 אנשים (דומה לצעד הראשון של אלגוריתם סלפרידג'-קונווי):
- 1. עמי חותך את העוגה לשלושה חלקים שווים בעיניו.
- 2. תמי מקצצת את הפרוסה הטובה ביותר בעיניה, כך שיהיו לה שתי פרוסות טובות ביותר.
- 3. השחקנים בוחרים פרוסה מבין הפרוסות הנמצאות על השולחן, לפי הסדר: רמי – תמי – עמי.

שימו לב – באלגוריתם זה לא כל העוגה מחולקת – יש שארית הנשארת על השולחן.

א. הוכיחו שהחלוקה המתקבלת היא ללא קנאה, וכל שחקן מקבל פרוסה השווה בעיניו לפחות **1/4** מהשווי הכללי של העוגה.
ב. עכשיו נניח שצריך לחלק עוגה בין 4 שחקנים. תארו אלגוריתם לחלוקה עם שארית, הנותן חלוקה ללא קנאה, שבה כל שחקן מקבל פרוסה השווה בעיניו לפחות **1/8** מהשווי הכללי של העוגה.

תשובה 6:
א. רמי בוחר ראשון, יש לפניו 4 חתיכות, הוא יבחר את החתיכה הכי שווה בעניו, לכן שווייה בעיניו לפחות ¼. תמי בוחרת שניה, תמי חילקה את החלק הכי שווה בעיניה, כך שיהיו שתי פרוסות טובות ביותר, ולכן לא משנה מה רמי בחר נשארה לפחות פרוסה אחת טובה ביותר בעיני תמי ולכן שוויה בעני תמי לפחות ¼. האחרון שבוחר חתיכה הוא עמי, עמי חילק בהתחלה את העוגה לשלושה חתיכות שוות, נניח רמי ותמי בוחרים את החתיכות שקוצצו על ידי תמי, נותר לעמי לבחור בין שני חתיכות השוות בעיניו 1/3 ולכן בהכרח לפחות ¼. נניח, רמי בוחר בחתיכה שלו קוצצה, נותר לעמי חתיכה שווה בעינוי 1/3 ולכן בהכרח לפחות ¼.

ב. חלוקת עוגה בין 4 אנשים(עמי, תמי,רמי וצומי), אלגוריתם:
1. עמי חותך את העוגה ל-4 חלקים השווים בעיניו.

2. תמי מקצצת את הפרוסה הטובה ביותר בעיניה, כך שיהיו שתי פרוסות טובות ביותר בעיניה.
3. רמי מקצץ את הפרוסה הטובה ביותר בעיניו, כך שיהיו שתי פרוסות טובות ביותר בעינוי.
4. השחקנים בוחרים פרוסה מבין הפרוסות הנמצאות על השולחן, לפי הסדר: צומי–רמי–תמי–עמי.

שאלה 7: **תיכנות – חלוקת-עוגה פרופורציונלית**
נניח שרוצים לחלק נהר חד-ממדי, המיוצג ע"י הקטע [0,1]. כל משתתף מיוצג ע"י המחלקה הבאה:

```
class Agent {
	double eval(double x);
	 // INPUT:   x, a number between 0 and 1.
	 // OUTPUT:  v, the value of [0,x]

	double mark(double v);
	 // INPUT:   v, a number between 0 and 1.
	 // OUTPUT:  a number such that the value of [0,x] is v.
}
```

א. כיתבו בשפה לבחירתכם (אפשר גם פסאודו-קוד) את אלגוריתם "חתוך ובחר":

```
void cutAndChoose(Agent a, Agent b)
```

הפונקציה מקבלת שני שחקנים וכותבת את החלוקה בפורמט הבא (לדוגמה):
Agent a receives [0,0.3]. Agent b receives [0.3,1].

ב. כיתבו בשפה לבחירתכם את אלגוריתם "המפחית האחרון":

```
void lastDiminisher(Agent[] agents)
```

הפונקציה מקבלת מערך של שחקנים וכותבת את החלוקה בפורמט הבא (לדוגמה):
Agent 0 receives [0,0.3]. Agent 1 receives [0.3,0.6]. Agent 2

<u>תשובה</u> <u>7</u>:

```
void cutAndChoose(Agent a, Agent b)
	double aPartition =  a.mark(1/2)
	if ( b.eval(aPartition) > ½) print (a receives [aPartition,1] , b receives [0,aPartition])
	else print (b receives [aPartition,1] , a receives [0,aPartition])
```

<u>חלוקה יעילה (שיעור 2):</u>

הגדרות:
שיפור פארטו: מצב א' נקרא שיפור פארטו של מצב ב', אם הוא טוב יותר לחלק מהמשתתפים, וטוב לפחות באותה מידה לכולם. "זה נהנה וזה לא חסר".
יעיל פארטו: מצב נקרא יעיל פארטו אם לא קיים מצב אחר שהוא שיפור פארטו שלו.

יעילות פארטו – תנאי הכרחי לבחירה שהיא "נכונה" מנקודת מבט כלכלית.

יעילות אלגוריתם "חתוך ובחר":
שאלה: האם האלגוריתמים לחלוקת עוגה שראינו הם מחזירים תמיד חלוקה שהיא יעילה פארטו?
תשובה: בדרך כלל אלגוריתם "חתוך ובחר" אינו יעיל פארטו.

משפט: אלגוריתם "חתוך ובחר מחזיר תשובה יעילה פארטו אם מתקיימים התנאים הבאים:
1. העוגה חד-ממדית.
2. שני השחקנים רוצים רק חתיכות קשירות.
3. לכל נקודה בעוגה יש ערך חיובי ממש.
4. החותך חותך לשני חלקים שווים ולא "מתחכם".

יעילות המקרה הכללי:
הנחות:
- ה"עוגה" מחולקת לאיזורים. הערך של כל שחקן אחיד בכל איזור.
- אין חשיבות לקשירות.

דרכים למציאת חלוקה יעילה פארטו:
1. דרך אחת למצוא חלוקה יעילה פארטו היא לפתור בעיית אופטימיזציה.
לדוגמה: מצא את החלוקה שבה סכום הערכים של השחקנים הוא מקסימלי:

$$\text{Maximize } V_1(X_1) + V_2(X_2)$$
$$\text{such that } (X_1, X_2) \text{ is a partition}$$

הבעיה הזאת ניתנת לפתרון בקלות במקרה של חלוקת עוגה: פשוט נותנים כל איזור מהעוגה לשחקן שעבורו הערך של האיזור הוא הכי גדול.

משפט: כל חלוקה הממקסמת את סכום הערכים היא יעילה פארטו.

אז מצאנו חלוקה יעילה אבל היא לא פרופורציונלית ויש בה קנאה.

2. דרך שניה, תהי F פונקציה כלשהי היא מונוטונית עולה:

$$\text{Maximize } F(V_1(X_1)) + F(V_2(X_2))$$
$$\text{such that } (X_1, X_2) \text{ is a partition}$$

הבעיה מאוד דומה לבעיה הקודמת רק כשהפעם לפני חישוב סכום הערכים, הפעלנו את F על כל ערך.
משפט: כל חלוקה הממקסמת פונקציה עולה כלשהי של סכום הערכם היא יעילה פרטו.

שאלה: איך פותרים בעיית אופטימיזציה כזאת?
תשובה: ישנם שתי דרכים. הראשונה, לכתוב פונקציה בכמה משתנים, לגזור אותה ולמצוא מקסימום. השניה, להשתמש בתוכנות מתמטיקה ידועות כמו: Mathematica.

3. דרך שלישית, מסתבר שיש פונקציה F, שאם ממקסמים את הסכום שלה, מקבלים חלוקה שהיא תמיד ללא קנאה (ולכן גם פרופורציונלית) – זוהי הפונקציה הלוגריתמית.
כלומר, פתרון הבעיה הבאה נותן חלוקה שהיא גם יעילה פארטו וגם ללא קנאה:

$$\text{Maximize sum all}-i\ \log(V_i(X_i))$$
$$\text{such that } (X_1, X_2, \ldots, X_n) \text{ is a partition}$$

משפט: כל חלוקה הממקסמת את סכום לוגי הערכים היא חלוקה ללא קנאה.

חלוקה קשירה, יעלה וללא קנאה:
ראינו שתמיד אפשר למצוא חלוקה שהיא:
- הוגנת ויעילה (מיקסום סכום הלוגים).
- הוגנת וקשירה (אלגוריתם "סו").
- יעילה וקשירה (תן הכל לבעל הערך הכי גבוהה).

שאלה: האם תמיד קיימת חלוקה הוגנת, יעילה וקשירה?
תשובה: לא! (באמצעות דוגמה נגדית).

<u>מטלה – חלוקה יעילה</u>

שאלה 1: מיקסום סכום של פונקציה עולה

נניח שאנחנו מחלקים "עוגה", המייצגת אוסף של משאבים, בעזרת בעיית האופטימיזציה הבאה:

Maximize $(V_1(X_1))^{0.6} + (V_2(X_2))^{0.6}$

such that (X_1, X_2) *is a partition*

(מציאת מקסימום של הסכום של הערכים בחזקת 0.6).

א. תנו דוגמה לבעיית-חלוקה שבה הפתרון הוא יעיל-פארטו וגם הוגן (ללא קנאה). תארו את הבעיה בטבלה, תארו את הפתרון, והוכיחו שהפתרון הוגן.

ב. תנו דוגמה לבעיית-חלוקה שבה הפתרון הוא יעיל-פארטו אבל לא הוגן (יש קנאה). תארו את הבעיה בטבלה, תארו את הפתרון, והוכיחו שהפתרון לא הוגן.

שאלה 2: חישוב חלוקה הוגנת ויעילה

עמי תמי וצומי רוצים להשתמש במחשב-העל המחלקתי לצורך ביצוע חישובים מורכבים. הערך של עמי הוא: 1*כמות הדיסק שהוא מקבל ועוד 2*כמות המעבד שהוא מקבל ועוד 3*כמות הזיכרון שהוא מקבל. הערכים של תמי ושל צומי נקבעים באופן דומה רק עם מספרים שונים, בהתאם לטבלה הבאה:

	זיכרון	**מעבד**	**דיסק**
עמי	3	2	1
תמי	6	5	4
צומי	9	8	7

כיתבו פקודה בשפת Mathematica (או בשפה אחרת לבחירתכם) המוצאת חלוקה יעילה-פארטו וללא קנאה של משאבי המיחשוב.

תשובה 2:

Find maximum $\{\log(1x+ 2y+ 3z)+ \log(4x_1+ 5y_1+ 6z_1)+ \log[7(1-x-x_1)+ 8(1-y-y_1)+ 9(1-z-z_1)]\}$

$0 <= x, y, z, x_1, y_1, z_1, 1-x-x_1, 1-y-y_1, 1-z-z_1 >= 1$

שאלה 3: יעילות-פארטו חלשה וחזקה

הגדרות:

מצב א נקרא **שיפור פארטו חזק** של מצב ב, אם מצב א טוב יותר *לכל* המשתתפים.

מצב נקרא **יעיל פארטו חלש** אם לא קיים מצב אחר שהוא שיפור-פארטו-חזק שלו.

א. תנו דוגמה לחלוקה שהיא יעילה-פארטו-חלש אבל לא יעילה-פארטו.

ב. הוכיחו שכל חלוקה יעילה-פארטו היא גם יעילה-פארטו-חלש.

ג. הוכיחו, שאם פונקציות הערך של כל השחקנים הן חיוביות ממש בכל נקודה ונקודה בעוגה, אז כל חלוקה יעילה-פארטו-חלש היא גם יעילה-פארטו.

ד. הוכיחו, שאם לכל השחקנים ישנה אותה פונקציית-ערך, אז כל החלוקות הן יעילות-פארטו.

תשובה 3:
א.

מסעדה:	א'	ב'
עמי	1	1
תמי	2	3

בחירה במסעדה א', היא יעילה פארטו חלשה. אבל היא לא יעילה פארטו יכלנו לבחור במסעדה ב' עמי לא היה מפסיד ותמי הייתה מרוויחה.

ב. תהיה בחירה כלשהי, נסמסנה ב-א', מהנתון בחירה א' היא יעלה פארטו. אזי לא קיים לה שיפור פארטו. מכאן שבהכרח לא קיים לה שיפור פארטו חזק, ולכן היא גם יעילה פארטו חלש.
הסבר: אם לא קיים ל-א' בחירה שמטיבה עם חלק מהשחקנים והשאר לא מפסידים בהכרח אין לה בחירה שבה כולם מרוויחים.

ד. מכיוון שלכל השחקנים יש את אותה פונקציית ערך, בכל חלוקה סכום הערכים יהיה זהה, מכיוון שחלוקה ממקסמת את סכום הערכים היא יעילה פארטו. לכל חלוקה לא קיים שיפור פארטו ולכן כל החלוקות הן יעילות פארטו (אותו סכום).

שאלה 4: **מיקסום סכום הערכים תחת אילוצי הגינות**
נתונה הבעיה:

Maximize $V_1(X_1) + V_2(X_2)$
such that (X_1, X_2) *is a partition*
and $V_1(X_1) \geq 1/2$ *and* $V_2(X_2) \geq 1/2$

א. הוכיחו שהפתרון לבעיה הוא תמיד חלוקה פרופורציונלית.
ב. הוכיחו שהפתרון לבעיה הוא תמיד חלוקה יעילה-פארטו.
ג. הוכיחו שהפתרון לבעיה הוא תמיד חלוקה ללא-קנאה.

תשובה 4:
א. האילוץ האחרון בדיוק עונה על ההגדרה של חלוקה פרופורציונלית, השחקן הראשון מעריך את החלק שלו בלפחות חצי, השחקן השני גם מעריך את החלק שלו בלפחות חצי, לכן החלוקה פרופורציונלית.

ב. האילוץ הראשון עונה על ההגדרה של חלוקה יעילה פרטו, כפי שהוכחנו בכיתה חלוקה הממקסמת את הכום הערכים היא יעלה פרטו, אבל בבעיה הזאת יש אילוצים נוספים, ולכן ההוכחה טיפה שונה. נניח בשלילה שקיים שיפור פארטו, נסמנו ב-*Z*. אזי , אחד(לפחות) השחקנים מרוויח , בה"כ נניח כי השחקן הראשון מרוויח, אזי:
$V_1(Y_1) > V_1(X_1)$ && $V_2(Y_2) >= V_2(X_2)$
עדיין מתקיים האילוץ של הבעיה. אבל הסכום שקיבלנו יותר גדול – סתירה למקסימליות של X.

ג. אם כל שחקן מעריך את החצי שלו בלפחות חצי אזי הוא מעריך את החצי השני לכל היותר חצי, ולכן כל שחקן לא מקנא בשני(נכון עבור שני שחקנים).

שאלה 5: **תיכנות – יעילות פארטו**
מטרת האלגוריתם היא לבדוק האם בחירה מסויימת היא יעילה פארטו. ישנם כמה שחקנים הצריכים לבחור באפשרות אחת מתוך כמה אפשרויות. כל שחקן מייחס ערך מסויים לכל אחת מהאפשרויות, לפי המחלקה הבאה:

```
class Agent {
	double value(int option);
	 // INPUT:   the index of an option.
	 // OUTPUT:  the value of the option to the agent.
}
```

א. כיתבו אלגוריתם המקבל מערך של שחקנים ושתי אפשרויות (כל אפשרות מיוצגת ע"י מספר שלם), ובודק האם אפשרות 1 היא שיפור פארטו של אפשרות 2:

```
bool isParetoImprovement(Agent[] agents, int option1, int option2)
```

ב. כיתבו אלגוריתם המקבל מערך של שחקנים, אפשרות, ומערך של כל האפשרויות, ובודק האם האפשרות הנתונה יעילה פארטו:

```
bool isParetoOptimal(Agent[] agents, int option, int[] allOptions)
```

תשובה 5:

```
bool isParetoImprovement(Agent[] agents, int option1, int option2)
	boolean bigger  =  false
	for( I =  0 to agents.length)
		if(agents[i].value(option1) < agents[i].value(option2)) return false
		if( agents[i].value(option1) > agents[i].value(option2) bigger =  true
	end-for
	if(bigger) return true
	return false
end-  isParetoImprovement
```

<u>חלוקת חפצים בדידים (שיעור 3):</u>

כשהחפצים בדידים, בדרך כלל אי אפשר למצוא חלוקה פרופורציונלית וללא קנאה. למשל: חלוקת בית – חלוקה הוגנת של שכר דירה.
פתרונות מקובלים:
1. הוספת כסף למערכת. דוגמה: אלגוריתמי חלוקת שכר דירה.
2. מציאת דרך יצירתית לחלק חפץ אחד. דוגמה: אלגוריתם "WIN-WIN" לגישור.
3. חלוקה ללא קנאה בקירוב. דוגמה: חלוקת תכשיטים ומקומות בקורסים.

חלוקת שכר דירה:
נתונים:
- דירה עם N חדרים ודמי שכירות נתונים P.
- N שותפים שרוצים לשכור יחד את הדירה.

האתגר הוא להחליט לגבי כל שותף:
- כמה כסף הוא ישלם. כאשר סכום הדיירים צריך להיות P.
- איזה חדר יקבל. צריך שלא תהיה קנאה.

ישנם שתי דרכים לאפיין העדפות של בני אדם:
- העדפות קרדינליות: אנשים יודעים לתת ערך מספרי לכל חפץ.
- העדפות אורדינליות: אנשים יודעים להגיד מה הם מעדיפים, אבל לא להגיד מספרים.

חלוקת שכר דירה – מודל אורדינלי:
הנחות:
- "חדרים סבירים": בכל חלוקה של שכר הדירה כל שוכר מוכן לקבל חדר כלשהו.
- "דיירים עניים": כל שוכר מעדיף חדר בחינם על פני חדר בתשלום.

פתרון עבור שני שוכרים (נשתמש באלגוריתם "חתוך ובחר"):
- אחד מחלק את שכר הדירה.
- השני בוחר חדר.

הכללה ל N שוכרים (אלגוריתם "סימונס Su, 1999"):
נניח ששכר הדירה הוא 1. כל תימחור של החדרים שקול ל-N מספרים שסכומם הוא 1. ולכן, אוסף כל הוקטורים האלה הוא סימפלקס. נבצע מישלוש של הסימפלקס – נחלק אותו לסימפלקסונים קטנים, נניח בגודל של אגורה אחת. עבור כל וקטור מחיר שנמצא על הקודקוד נשאל כל שותף "איזה חדר אתה מעדיף?". לפי הנחת "הדיירים העניים", כל דייר מעדיף בכל חלוקה את אחת הפרוסות הריקות (אחד החדרים שמחירם הוא 0). לפי הלמה של ספרנר, קיים משולשון שבו כל דייר מעדיף חדר אחר. זוהי חלוקה ללא קנאה בקירוב של אגורה!.

הבעיה במודל האורדינלי:
הנחת "הדיירים העניים" לא תמיד מתקיימת. לדוגמה: אם המרתף התחוב בחינם, והחדר הגדול הפונה אל הים עולה שקל אחד, רוב האנשים יעדיפו לשלם עבור החדר.

חלוקת שכר דירה – מודל קרדינלי:
במודל הקרדינלי, כל אחד מהשותפים מייחס ערך מספרי לכל חדר, המשקף את הסכום המירבי שהוא מוכן לשלם עבור החדר.
הנחות:
– "החדרים סבירים": כל דייר מייחס ערך כספי לכל חדר, סכום הערכים גדול שווה למחיר הדירה.
– "קוואזי-לינאריות": התועלת של דייר היא, ערך החדר פחות המחיר שלו.
– נשים לב כי הנחת "הדיירים העניים" בדרך כלל לא מתקיימת פה, סותר את ה"קוואזי-לינאריות".

במודל זה חלוקה ללא קנאה היא חלקה שבה כל שותף מקבל חדר עם תג מחיר, כך שהתועלת של כל שותף מהחדר שלו גדולה לפחות כמו התועלת שלו מכל חדר אחר.

משפט: בכל חלוקה ללא קנאה, סכום הערכים של הדיירים בחדרים שהם גרים הוא מקסימלי.

מסקנות:
1. כל השמת חדרים ללא קנאה היא יעילה פארטו.
2. כדי למצוא חלוקת שכר דירה ללא קנאה, צריך אלגוריתם להשמה הממקסמת את סכום הערכים.

המשפט הזה נותן לנו כיוון לאלגוריתם למציאת חלוקה ללא קנאה:
שלב א: נמצא השמה של אנשים לחדרים, הממקסמת את סכום הערכים.
שלב ב: נקבע מחיר לכל חדר, כך שההשמה עם המחירים תהיה ללא קנאה.

שלב א, מציאת השמה עם סכום ערכים מקסימלי:
בעיית מיקסום סכום הערכים ידועה בשמות שונים: בעיית ההשמה, שידוך עם משקל מקסימלי.

אנחנו נראה איך להפוך את הבעיה לבעיית אופטימיזציה שאפשר לפתור ב Mathematica:
– לכל קשת בגרף (בין i-j), יהיה משתנה x[i,j]: שערכו יהיה 1 אם הקשת נמצאת בשידוך, אחת ערכו יהיה 0.

For all-i: Sum-j x[i,j]= 1; For all-j: Sum-i x[i,j]= 1

המשקל הכולל של השידוך: Sum-i-j w[i,j] * x[i,j]

לסיכום, זו התוכנית שיש לפתור:

$$\text{Maximize} \quad \text{Sum}_{i,j}\ w[i,j] * x[i,j]$$

$$\text{Such that For all-i: Sum}_j\ x[i,j]= 1$$
$$\text{For all-j: Sum}_i\ x[i,j]= 1$$

$$\text{For all-i,j: } 1>= x[i,j] >= 0$$
$$\text{For all-i,j: } x[i,j] \text{ in } Z$$

הבעיה היחידה היא האילוץ האחרון שאומר שכל המספרים צריכים להיות שלמים. בעיית אופטימיזציה עם מספרים שלמים היא בעיה NP-קשה!

משפט: בבעיית מציאת שידוך מקסימלי בגרף דו-צדדי, אם קיים פתרון אופטימלי עם משתנים לא שלמים, אז קיים פתרון אופטימלי שבו כל המשתנים שלמים.

שלב ב: קביעת המחירים:

מצאנו השמה הממקסמת את הערכים. כעת, צריך לקבוע מחירים על מנת שההשמה תהיה ללא קנאה, וסכום המחירים יהיה שווה לשכר הדירה. נעשה זאת באמצעות בעיית אופטימיזציה:

- לכל חדר-i יהיה משתנה p[i] הקבוע את מחיר החדר.
- d[i] מציין את הדייר המשויך לחדר מספר-i.

Minimize Sum_i p[i]
Such that For all-i,j: w[d[i],i]-p[i] >= w[d[i],j] – p[j]

- במקרה של גירעון/עודף מחלקים את העודף / גירעון שווה בשווה בין כולם.
- קיבלנו חלוקה ללא קנאה!

בעיית הטרמפיסט:

משפט: במודל הקרדינלי, ייתכן שבכל חלוקה ללא קנאה, אחד הדיירים ישלם מחיר שלילי (צריך לשלם לו כדי שיסכים לגור איתנו).

	עובד רק עם "דיירים עניים"	קנאה	דיירים שמקבלים כסף
אלגוריתם סו והמשולשים	כן	לא	לא
האלגוריתם ההונגרי ודומיו	לא	לא	כן
אלגוריתם הונגרי עם מחיר מינ. 0	לא	כן	לא

מטלה – חלוקת שכר-דירה

שאלה 1: אלגוריתם חלופי לחלוקת שכר-דירה

פתחו אלגוריתם חלופי לחלוקת חדרים ושכר-דירה בין שלושה שותפים.
בחרו אחד מהרעיונות הבאים (שהעליתם בהרצאה):
א. התאמה של אלגוריתם Selfridge–Conway לחלוקת-עוגה ללא קנאה.
ב. ביצוע מכרז על כל חדר.
ג. מיקסום סכום המנות (ערך חלקי מחיר).
ד. רעיון אחר כלשהו.
יש להוכיח פורמלית שהאלגוריתם שלכם עובד. אם ניסיתם ונתקעתם – נסו כיוון אחר. אם שוב נתקעתם – נסו כיוון אחר. אם שוב נתקעתם – הסבירו בפירוט למה נתקעתם (למה הרעיון לא עובד).

תשובה 1:

א. עמי, תמי וצומי רוצים לשכור יחדיו דירה.
1. עמי מחלק את המחירים לחדרים לפי ראות עיניו, שיהיו שווים.
2. תמי תבחר את החדר הכי טוב בעיניה, ותוסיף לו מחיר ככה שהוא יהיה שווה לחדר השני בטיבו.
3. בוחרים חדרים לפי הסדר: צומי-תמי-עמי, תמי חייבת לבחור בחדר שהעלתה לו מחיר ומידה והחדר לא נבחר קודם ע"י צומי.
4. במידה ויש גירעון או יתרה מתחלקים בו שווה ביניהם.

נוכיח כי האלגוריתם מבצע חלוקה ללא קנאה ולכן הוא ממקסם את סכום הערכים ולכן הוא יעיל פארטו. עמי מחלק את המחירים לחדרים לפי ראות עיניו כך שהם יהיו שווים. תמי תקח את החדר הכי טוב בעיניה ותוסיף לו מחיר ככה שיהיה שווה לחדר השני בטיבו, כעת יש בעיני תמי שני חדרים הכי טובים. ואז בוחרים לפי הסדר, צומי בוחר באיזה חדר שבא לו, הוא לא מקנא באף אחד כי הוא בחר בחדר הכי טוב מבחינתו ואף אחד לא מקנא בו, כי יש עוד לפחות חדר אחד הכי טוב בעני תמי ועמי. שניה בוחרת תמי, במידה וצומי בחר בחדר שהיא הוסיפה לו מחיר היא תבחר בחדר השני הכי טוב בעיניה, אחרת היא תבחר בחדר הכי טוב בעיניה, החדר שהוסיפה לו מחיר. בכל מקרה תמי לא מקנא באף אחד כי היא לקחה את החדר הכי טוב בעיניה וצומי לא מקנא בה כי בחר בחדר הכי טוב בעיניו, ועמי לא מקנא בה כי הוא חילק את החדרים ויש עדיין חדר שהוא הכי טוב בעיניו. לאחר מכן עמי מקבל את החדר הנותר הוא לא מקנא באף אחד כי הוא חילק בהתחלה את המחירים כראות עיניו ובהכרח נשאר חדר "טוב" בעיניו. ואף אחד לא מקנא בו כי הם בחרו לפניו חדר הכי טוב בעיניהם.

שאלה 2: בעיית תחנת המוניות

בתחנת מוניות עובדים שלושה נהגים. כל נהג נמצא עכשיו במקום אחר. התחנה מקבלת בו-זמנית שלוש פניות מנוסעים הנמצאים במקומות שונים. התחנה צריכה להחליט איזה נהג לשלוח לאיזה נוסע. הנתונים:

- המרחק בין נהג i לבין נוסע j הוא: d[i,j] (ק"מ).
- המרחק בין נוסע j לבין היעד שלו הוא x[j] (ק"מ).
- מחיר של דלק לקילומטר הוא p (ש"ח).

כיתבו פקודה בשפת Mathematica (או שפה דומה) שתעזור לתחנה למצוא את ההשמה הזולה ביותר של נהגים לנוסעים.

תשובה 2:
נבנה גרף דו-צדדי, עם משקלים על הצלעות, כך שכל משקל על הצלע מהווה את המרחק בין הנהג לנוסע ועוד המרחק בין הנוסע ליעד שלו. נגדיר: $z[i,j]= 1$ אם הקשת היא חלק מהשידוך האופטימלי, אחרת, 0. כעת נצטרך לפתור את בעיית האופטימיזציה הבאה:

Minimize $Sum_{i,j} (d[i,j] + x[j]) * z[i,j]$

Such that $Sum_j z[i,j] = 1$, $Sum_i z[i,j] = 1$

For all $0<= z[i,j] <= 1$

For all $z[i,j]$ in Z

שאלה 3: **מחיר אגליטרי**
בכל השמה של דיירים לחדרים, ישנן דרכים רבות לקבוע את מחירי החדרים. בכל וקטור מחירים, התועלת של כל דייר היא ערך החדר שקיבל עבורו פחות מחיר החדר. בכל וקטור מחירים, יש דייר אחד (או יותר) שהתועלת שלו הכי קטנה. וקטור-המחירים ה**אגליטרי** (egalitarian) היא הוקטור שבו התועלת הקטנה ביותר היא גדולה יותר מכל שאר הוקטורים.
א. הדגימו את המושג "וקטור-מחירים אגליטרי" על בעיית שכר-דירה עם שני חדרים ושני דיירים: המציאו ערכים, מיצאו את ההשמה הממקסמת את סכום הערכים, והראו שני וקטורי-מחירים – אחד אגליטרי ואחד לא אגליטרי.
ב. תנו דוגמה עם שלושה חדרים ושלושה דיירים, שבה בוקטור-המחירים האגליטרי יש קנאה.
ג. תנו דוגמה עם שלושה חדרים ושלושה דיירים, שבה בוקטור-המחירים האגליטרי אין קנאה.

תשובה 3:
א. המחיר הכולל של הדירה הוא 1000

חדרים:	חדר א'	חדר ב'
עמי	900	200
תמי	800	300

מחיר אגליטרי:
עמי- חדר א' במחיר 800, הנאה 100.
תמי- חדר ב' במחיר 200, הנאה 100.

מחיר לא אגליטרי:
עמי- חדר א' במחיר 900, הנאה 0.
תמי- בדר ב' במחיר 100, הנאה 200.

ב. מחיר דירה כולל 100

חדרים:	חדר א'	חדר ב'	חדר ג'
עמי	40	80	10
תמי	10	70	20
צומי	30	60	10

מחיר אגליטרי:
עמי– חדר ב' במחיר 70, הנאה 10.
תמי– חדר ג' במחיר 10, הנאה 10.
צומי– חדר א' במחיר 20, הנאה 10.
עמי מקנא בצומי, אם היה מקבל את חדר א' ב 20 ההנאה שלו הייתה 20.

ג. מחיר דירה כולל 100

חדרים:	חדר א'	חדר ב'	חדר ג'
עמי	50	0	0
תמי	0	50	0
צומי	0	0	30

מחיר אגליטרי:
עמי– חדר א' במחיר 40, הנאה 10.
תמי– חדר ב' במחיר 40, הנאה 10.
צומי– חדר ג' במחיר 20, הנאה 10.
המחיר אגליטרי ואין קנאה.

שאלה 4: סכום מחירים מקסימלי
נניח שמצאנו השמה הממקסמת את סכום הערכים, ואנחנו רוצים למצוא מחיר לכל חדר. במקום למצוא וקטור-מחירים עם סכום מינימלי, אפשר לנסות למצוא וקטור-מחירים עם סכום מקסימלי:

Maximize $\quad \text{Sum}_i\ p[i]$

Such that $\quad$ For all i, j: $w[d[i],\ i] - p[i] \geq w[d[i],\ j] - p[j]$

א. מה הבעיה בתוכנית זו? (אפשר לפתור סעיף זה בשתי דרכים: להריץ ב Mathematica ולראות איזו הודעת שגיאה-מתקבלת, או לנסות לפתור את הבעיה ידנית במקרים פרטיים ולזהות את הבעיה).
ב. איזה אילוץ אפשר להוסיף לתוכנית, כך שיתקבל פתרון הגיוני?

תשובה 4:
א. הבעיה בתוכנית הזאת היא, שהיא בעיה שאינה חסומה, לכן החישוב לא יסתיים. מכיוון שעל כל חלוקה כלשהי ללא קנאה נוכל להוסיף לכולם אותו מחיר והחלוקה תשאר ללא קנאה.
ב. נוסיף אילוץ הגיוני שיחסום את הבעיה:

For all–i: $p[i] <= w[d[i],i]$

כלומר, המחיר של כל דייר עבור החדר לא יעלה מעבר לערך שלו עבור החדר.

שאלה 5: חלוקת שכר-דירה – אלגוריתם spliddit
בעזרת אתר spliddit, חפשו בעיות חלוקת שכר-דירה עם שלושה שותפים ושכר-דירה חיובי, עם התכונות הבאות:
א. כל השותפים משלמים בדיוק שליש משכר-הדירה;
ב. שותף אחד משלם אפס, ושני השותפים האחרים משלמים בדיוק חצי משכר-הדירה;
ג. שני שותפים משלמים אפס, והשותף השלישי משלם את כל שכר-הדירה;
ד. אחד השותפים משלם מחיר שלילי.

שאלה 6: תיכנות – חלוקת שכר-דירה ללא קנאה

רוצים לחלק n חדרים ל-n דיירים. כל דייר מיוצג ע"י המחלקה הבאה:

```
class Agent {
	int bestRoom(int[] prices);
    // INPUT:   the prices of the n rooms, in shekels.
    // OUTPUT:  the index of a room that the agent most prefers in these prices. Index is between 0 and n-1.
};
```

כיתבו בשפה לבחירתכם, או בפסאודו-קוד, אלגוריתם המקבל כקלט n שחקנים ואת מחיר הדירה הכולל בשקלים, ומוצא השמת-חדרים ללא קנאה עד כדי שקל אחד. אם זה מקל עליכם – אפשר להניח ש n= 3.

כותרת הפונקציה:

```
void findEnvyFreeAssignment(Agent[] agents, int totalRent)
```

פלט לדוגמה:

Agent 0 receives room 2 for 163 shekels.
Agent 1 receives room 1 for 274 shekels.
Agent 2 receives room 0 for 343 shekels.

תשובה 6:

מכיוון שאין לנו נתונים עבור הערך של כל דייר עבור חדר מסויים, אלא רק בהינתן ווקטור מחירים מסויים מהי העדפת הדיירים – העדפה אורדינלית. בהנחה שהעדפות מקיימות את "הדיירים העניים", "החדרים הם סבירים". ולכן נעבור על כל התמורות האפשריות בווקטור המחירים וננסה למצוא תמורה שבה כל דייר יעדיף חדר אחר – תמורה זו תהווה חלוקה ללא קנאה. (בקוד, מוצאים השמה ללא קנאה ממש, לא בקירוב. ע"מ למצוא קירוב של שקל צריך לבדוק עבור כל מחיר +- שקל עם כל שאר המחירים).

```
void findEnvyFreeAssignment(Agent[] agents, int totalRent)
	for(from price1 = 0 to totalRent)
		for(from price2 = 0 to totalRent-price1)
			price3= totalRent-price1-price2
			prices[] =  {price1,price2,price3}
			index1= agents[0].bestRoom(prices)
			index2= agents[1].bestRoom(prices)
			index3= agents[2].bestRoom(prices)
			if( (index1 !=  index2) and (index1 !=  index3) and (index2 !=  index3) )
				print( Agent 0 receives room index1  for prices[index1] shekels.
				       Agent 1 receives room index2  for prices[index2] shekels.
				       Agent 2 receives room index3  for prices[index3] shekels.)
				break function.
			end-if
		end-for
	end-for
end-findEnvyFreeAssignment
```

חלוקה הוגנת ביישוב סכסוכים (שיעור 4):
אלגוריתם לחלוקה הוגנת, המשמש לא רק לחלוקה של חפצים אלא גם לחלוקה של נושאים שיש עליהם מחלוקת, כמו למשל במשפטי גירושין או פירוק שותפויות. האלגוריתם יכול לשמש לגישור ולמציאת פתרון שיהיה טוב לשני הצדדים ולכן הוא נקרא "the win–win solution" או "adjusted winner" (המנתח המתוקן).
נתונים:
– שני שותפים.
– M חפצים או נושאים שיש עליהם מחלוקת.
– כל שותף מייחס ערך/אחוזים לכל נושא.

האתגר הוא להחליט מי יקבל כל חפץ/נושא כך ש:
– לא תהיה קנאה.
– התוצאה תהיה יעילה פארטו.
– נצטרך לחתוך חפץ אחד לכל היותר.

אלגוריתם "המנצח המתוקן" 1996:
– סדר את החפצים בסדר עולה של היחס בין שני המשתתפים.
– אתחול: תן את כל החפצים לאחד המשתתפים.
– העבר חפצים למשתתף השני, עד ש:
  – הסכום של המשתתפים שווה.
  – יש חפץ אחד שאם נחתוך אותו הסכום ישתווה.

משפט: אלגוריתם המנצח המתוקן מחזיר תמיד חלוקה יעילה פארטו.
משפט: אלגוריתם המנצח המתוקן מחזיר תמיד חלוקה ללא קנאה.

חלוקה הוגנת בקירוב:
חלוקה הוגנת של חפצים בדידים שאי אפשר לחתוך, כמו בתים, תכשיטים.

הגדרה: חלוקה נקראת "ללא קנאה מלבד 1" (EF1) אם לכל שני משתתפים א' ו–ב', קיים חפץ כלשהו, שאם נוריד מהסל של ב', אז שחקן א' לא יקנא בו.

המשמעות: רמת הקנאה היא הקטנה ביותר האפשרית, בהתחשבות שהחפצים בדידים.

אלגוריתם "מעגלי הקנאה":
עוברים על החפצים בסדר שרירותי. לכל חפץ:
1. נותנים את החפץ לשחקן שאף אחד לא מקנא בו.
2. אם אין כזה – סימן שיש מעגל קנאה. מחליפים סלים בניגוד לכיוון הקנאה.
מבצעים את 2 עד שאין מעגלים, ואז חוזרים ל 1.

משפט: אם יש M חפצים ו–N שחקנים, אז זמן הריצה של אלגוריתם מעגלי הקנאה הוא $O(MN^3)$.
משפט: האלגוריתם הנ"ל מחזיר חלוקה EF1.
משפט: אלגוריתם מעגלי הקנאה עלול להחזיר תוצאה שאינה יעילה פארטו.

כשה"עוגה" רציפה תמיד קיימת חלוקה EF ויעילה.(חלוקה הממקסמת את מכפלת ערכי המשתתפים).
שאלה: האם כשהחפצים בדידים קיימת חלוקה EF1 ויעילה ?
תשובה: כן! התגלה בשנת 2016.

משפט: נניח ש:
– ההעדפות אדיטיביות: ערך של סל הוא סכום הערכים של החפצים בסל.
– קיימת לפחות חלוקה אחת שבה כל שחקן מקבל ערך גדול מ 0.
אז, כל חלוקה הממקסמת את מכפלת ערכי המשתתפים היא גם יעילה פארטו וגם EF1.

איך מוצאים חלוקה כזאת ?

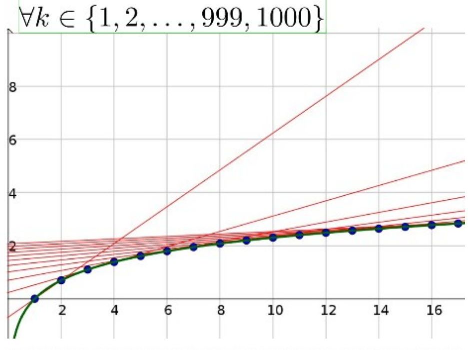
נסמן:
– $V_i(g)$ = the value of good g to player I
– $X_{i,g}$ = quantity of good g given to player I.
אנחנו רוצים לפתור את הבעיה:

$$\max \ \Sigma^{n}_{i=1} \log\left(\Sigma^{m}_{g=1} X_{i,g} * V_i(g)\right)$$
$$\text{such that For all } g: \ \Sigma^{n}_{i=1} X_{i,g} = 1$$

כאשר $X_{i,g}$ רציפים זה קל. כאשר הם בדידים זה קשה!

הבעיה המקורית היא קמורה אבל לא לינראית ולכן היא קשה.
נבצע טריק על מנת להפוך את הבעיה ללינארית: נניח שכל הערכים הם בין 1 ל 1000. נחליף את האילוץ האמצעי ב-1000 אילוצים לינאריים.
הרעיון: מחליפים את הפונקציה הלוגריתמית באוסף של 1000 פונקציות לינאריות שחוסמות אותה מלמעלה.
פתרון אופטימלי לבעיה הלינארית הוא גם פתרון אופטימלי לבעיה המקורית!

$$W_i \le \log k + [\log(k+1) - \log k] \cdot \left[\sum_{g=1}^{m} x_{i,g} \cdot V_i(g) - k\right]$$
$$\forall k \in \{1, 2, \ldots, 999, 1000\}$$



<u>**מטלה – חלוקת חפצים בדידים**</u>

שאלה 1: **חלוקה ללא קנאה מלבד החפץ הכי גרוע**
הגדרה: חלוקה של חפצים נקראת "ללא קנאה מלבד החפץ הכי גרוע" (בקיצור EFX), אם לכל שני שחקנים א, ב, **ולכל** חפץ שנוריד מהסל של ב, שחקן א לא יקנא בשחקן ב.
א. הראו שהתנאי EFX הוא ממש חזק יותר מהתנאי EF1 (תנו דוגמה לחלוקה EF1 שהיא לא EFX).
ב. יש שני שחקנים עם העדפות אדיטיביות (= לכל שחקן, ערך של סל הוא סכום הערכים של החפצים בסל), וזהות (= לשני השחקנים יש אותם ערכים לכל החפצים). תארו אלגוריתם המוצא חלוקה EFX.
ג. יש שני שחקנים עם העדפות אדיטיביות, אבל לא בהכרח זהות. תארו אלגוריתם המוצא חלוקה EFX.

תשובה 2:
א.

	יהלום	זהב	ברקת
תמי	10	20	70
עמי	10	20	70

החלוקה:
תמי מקבלת: יהלום, ברקת.
עמי מקבלת: זהב.
עמי מקנא בתמי רק וקיים חפץ שאם נעביר אותו לעמי (ברקת) אז הוא לא יקנא ולכן EF1.
לעומת זאת החלוקה היא לא EFX כי לא לכל חפץ שנוריד מתמי עמי יפסיק לקנא, נניח אם יהלום עמי עדיין יקנא.

ב. נשתמש באלגוריתם "מעגלי הקנאה" עם שינויי קל, נסדר את החפצים בסדר יורד של הערך , ומחלקים את החפצים מהגדול לקטן כך שלכל חפץ:
1. נותנים את החפץ לשחקן שאף אחד לא מקנא בו.
2. אם אין כזה – סימן שיש מעגל קנאה. מחליפים סלים בניגוד לכיוון הקנאה.
מבצעים את 2 עד שאין מעגלים, ואז חוזרים ל 1.
החלוקה היא ללא קנאה מבלד החפץ הכי גרוע. ולכן EFX.

ג. פתרון של אראל: אחד השחקנים מחלק את החפצים כאילו הוא שני שחקנים ("מכפיל" את עצמו), מחלק את החפצים לפי האלגוריתם של סעיף ב'. לאחר מכן השחקן השני בוחר את אחת הקבוצות והשחקן הראשון מקבל את הנותרת. השחקן שחילק מקנא EFX לפי סעיף ב', השחקן השני לא מקנא כי הוא בחר קודם בקבוצה הטובה בעיניו.

שאלה 2: **אלגוריתם מעגלי הקנאה – סדר החפצים**
כזכור, אלגוריתם מעגלי-הקנאה מתחיל בכך שהוא מסדר את החפצים בסדר שרירותי כלשהו.
כלומר, בכל סדר שבו נסדר את החפצים, אנחנו עשויים לקבל תוצאה שונה.
א. הראו דוגמה עם 3 אנשים ו-3 חפצים, שבה כל סידור של החפצים נותן תוצאה אחרת (סה"כ 6 תוצאות).
ב. הראו דוגמה עם 3 אנשים ו-3 חפצים, שבה כל סידור של החפצים נותן אותה תוצאה.
ג. הראו דוגמה עם 3 אנשים ו-3 חפצים, שבה מספר התוצאות האפשריות גדול מ-1 וקטן מ-6.

תשובה 2:
א.

	יהלום	זהב	ברקת
עמי	3	2	1
תמי	3	2	1
רמי	3	2	1

על כל סידור של חפצים נקבל תוצאה אחרת:
יהלום–זהב–ברקת: עמי יהלום, תמי זהב, רמי ברקת – אין מעגל קנאה,אין החלפות (אף אחד לא מקנא ברמי)
זהב–יהלום–ברקת: עמי זהב, תמי יהלום, רמי ברקת – אין מעגל קנאה,אין החלפות (אף אחד לא מקנא ברמי)
זהב–ברקת–יהלום: עמי זהב, תמי ברקת, רמי יהלום – אין מעגל קנאה,אין החלפות (אף אחד לא מקנא בתמי)
יהלום–ברקת–זהב: עמי יהלום, תמי ברקת, רמי זהב – אין מעגל קנאה,אין החלפות (אף אחד לא מקנא תמי)
ברקת–זהב–יהלום: עמי ברקת, תמי זהב, רמי יהלום – אין מעגל קנאה,אין החלפות (אף אחד לא מקנא בעמי)
ברקת–יהלום–זהב: עמי ברקת, תמי יהלום, רמי זהב – אין מעגל קנאה,אין החלפות (אף אחד לא מקנא בעמי)
ב.

	יהלום	זהב	ברקת
עמי	1	1	1
תמי	1	1	1
רמי	1	1	1

לשלושת האנשים אותה עדיפות לשלושת הפריטים על כל תמורה אפשרית של הפריטים, התוצאה תהיה זהה.
ג.

	יהלום	זהב	ברקת
עמי	3	1	1
תמי	3	1	1
רמי	3	1	1

שאלה 3: **מיקסום מכפלת הערכים – חפצים רציפים**
כזכור, למדנו שאלגוריתם מיקסום–מכפלת–הערכים נותן חלוקה שהיא ללא קנאה כאשר החפצים הם רציפים (ניתנים לחיתוך). תנו דוגמאות לחלוקה של 3 חפצים רציפים בין 2 אנשים, שבהן:
א. אלגוריתם מיקסום–מכפלת–הערכים לא צריך לחתוך אף חפץ.
ב. אלגוריתם מיקסום–מכפלת–הערכים צריך לחתוך חפץ אחד בדיוק.
ג. אלגוריתם מיקסום–מכפלת–הערכים צריך לחתוך שני חפצים בדיוק.
ד. אלגוריתם מיקסום–מכפלת–הערכים צריך לחתוך שלושה חפצים בדיוק.

תשובה 3:
א.

	דיסק	זיכרון	מעבד
עמי	1	1	1
תמי	0	0	0

ב.

	דיסק	זיכרון	מעבד
עמי	1	1	1
תמי	0	0	1

שאלה 4: **מיקסום מכפלת הערכים – חפצים בדידים**
כזכור, למדנו שאלגוריתם מיקסום-מכפלת-הערכים נותן חלוקה שהיא ללא-קנאה-מלבד-1 כאשר החפצים הם בדידים (לא ניתנים לחיתוך). תנו דוגמאות לחלוקה של 4 חפצים בדידים בין 2 אנשים, שבהן:
א. אלגוריתם מיקסום-מכפלת-הערכים מחזיר חלוקה שהיא ממש ללא קנאה.
ב. אלגוריתם מיקסום-מכפלת-הערכים נותן חפץ 1 לאדם אחד ו-3 חפצים לשני.
ג. אלגוריתם מיקסום-מכפלת-הערכים נותן נותן 2 חפצים לכל אחד.

תשובה 4:
א.

	יהלום	זהב	נחושת	ברקת
עמי	1	1	0	0
תמי	0	0	1	1

האלגוריתם יתן לעמי את היהלום והזהב ולתמי את הנחושת והברקת, אין קנאה ביניהם.
ב.

	יהלום	זהב	נחושת	ברקת
עמי	1	0	0	0
תמי	0	1	1	1

האלגוריתם יתן לעמי את היהלום ולתמי את הזהב, נחושת והברקת.
ג. זהה לטבלה של א

שאלה 5: **חלוקת תיקים בין מפלגות בקואליציה**
לאחר הבחירות, שתי מפלגות החליטו להקים ממשלה ביחד, אבל הן בזבזו חודש שלם בויכוחים על איזה מפלגה תקבל איזה תיק. לאחר שהתייאשו מהויכוחים, הן פנו אליכם כדי שתעזרו להם להחליט.
א. הציעו אלגוריתם שיאפשר לשתי המפלגות להחליט על חלוקת-תיקים הוגנת ויעילה-פארטו.
שימו לב: ההגינות צריכה להתייחס לגדלים השונים של המפלגות – מספר המנדטים שכל מפלגה קיבלה בבחירות. למפלגה גדולה יותר יש זכות לקבל יותר תיקים (או תיקים יותר "שווים").
ב. הוכיחו שהאלגוריתם מקיים שלוש תכונות: יעילות פארטו, הגינות (בהתאם לגדלים השונים), וחיתוך תיק אחד לכל היותר.

ג. הדגימו את פעולת האלגוריתם שלכם על אוסף התיקים שהיו באחת מממשלות ישראל האחרונות (לבחירתכם), ועל שתי מפלגות לבחירתכם.

תשובה 5:
א. פתרון של אראל: על בסיס האלגוריתם של "המנצח המתוקן":

- סדר את התיקים בסדר עולה של היחס בין שני המפלגות.
- אתחול: תן את כל התיקים למפלגה הגדולה.
- העבר תיקים למפלגה הקטנה, עד ש:
  - היחס בין כמות המנדטים לתיקים של כל מפלגה מופר.
- יש תיק אחד שאם נחלק אותו היחס יישמר.

שאלה 6: **בעיית ה-**Partition
א. קיראו על "בעיית החלוקה": https://en.wikipedia.org/wiki/Partition_problem.
ב. בחרו אחד מהאלגוריתמים לפתרון היוריסטי/מקורב של הבעיה וממשו אותו בשפת-תיכנות לפי בחירתכם.
ג. הדגימו את פעולת האלגוריתם על כמה קלטים לא טריביאליים.

שאלה 7: **תיכנות - חלוקה ללא-קנאה-בקירוב**
נתונה המחלקה הבאה:

```
class Agent {
	double item_value(int item_index);
}
```

המחלקה מייצגת שחקן המשתתף במשחק חלוקה הוגנת. יש בה פונקציה אחת המתארת את הערך שהשחקן מייחס לחפץ שהאינדקס שלו הוא item_index.
כיתבו פונקציה הבודקת האם חלוקת-חפצים נתונה כלשהי היא EFX (ראו שאלה 1 למעלה). כותרת הפונקציה:

```
bool is_EFX(Agent[] agents, set<int>[] bundles);
```

הפרמטר agents הוא מערך בגודל n המייצג את השחקנים.
הפרמטר bundles הוא מערך באותו גודל - n - המייצג את החלוקה: bundle[i] הוא אוסף אינדקסי החפצים שמקבל שחקן i.

מכרזים אמיתיים (שיעור 5):

מה מוכרים במכרזים:
– שלל של מלחמה.
– חפצי אומנות.
– משאבים ציבוריים (קרקעות, תדרים).
– עליה לתורה.
– פרסומות (מנועי חיפוש, רשתות חברתיות).

סוגי מכרזים:
– מכרז אנגלי: המשתתפים מעלים את המחיר, האחרון שנשאר זוכה.
– מכרז יפני: הכרוז מעלה את המחיר, האחרון שנשאר זוכה.
– מכרז הולנדי: הכרוז מוריד את המחיר, הראשון שמצטרף זוכה.
– מעטפות חתומות, מחיר ראשון (כל משתתף כותב מספר במעטפה ומגיש לכרוז. הכרוז פותח את כל המעטפות. מי שכתב את המחיר הגבוהה ביותר זוכה, ומשלם את המחיר שהוא הכריז) .
– מעטפות חתומות, מחיר שני.(כל משתתף כותב מספר במעטפה ומגיש לכרוז. הכרוז פותח את כל המעטפות. מי שכתב את המחיר הגבוהה ביותר זוכה, ומשלם את המחיר השני בגודלו).

מכרז אמיתי: מכרז הוא אמיתי אם לכל משתתף j כדאי להכריז את ערכו האמיתי v_j, לא משנה מה יעשו האחרים.

המשמעות: קל יותר למשתתפים, אין צורך "לרגל" אחרי משתתפים אחרים.

משפט: מכרז מחיר ראשון אינו אמיתי.
אינטואיציה: התועלת של הזוכה היא 0, וגם התועלת של האנשים שלא זכו היא 0. לכן הזוכה יכול לנסות להוריד במעט את ההצעה שלו מתחת לערך האמיתי שלו, לזכות, ולשפר את התועלת.

מכרז מחיר שני – מכרז ויקרי:
הגדרה: מכרז ויקרי הוא:
– המשתתפים כותבית את ההכרזות במטעפות.
– המעטפות נפתחות ומסודרות בסדר יורד.
– בעל ההכרזה הגבוהה ביותר זוכה בחפץ.
– הזוכה משלם את ההכרזה השניה.

משפט: מכרז ויקרי הוא אמיתי.
משפט: מכרז ויקרי הוא יעיל פארטו.

מכרז פרסום:
יש כמה חפצים למכירה, כל אחד באיכות שונה.
הנחות:
לכל משבצת k יש הסתברות הקלקה r_k.

$$r_1 > r_2 > r_3 > \dots \dots r_n$$

לכל מפרסם j יש ערך הקלקה v_j.(המחושב על ידי סטטיסטיקות שעושה המפרסם).
מכאן כל מפרסם מעריך את משבצת k כ: $r_k * v_j$.

המטרה שלנו היא למצוא מכרז שהוא יעיל פארטו ואמיתי.
משפט: הקצאת מקומות למפרסמים היא יעילה פארטו אם ורק אם היא ממקסמת את סכום הערכים:

$$v_1 * r_{k(1)} + v_2 * r_{k(2)} + \dots ..$$

נשתמש באלגוריתם חמדני: סדר את המפרסמים בסדר יורד של v_j:

$$v_1 > v_2 > \dots$$

תן למפרסם ה j את המקום ה j.

משפט: האלגוריתם המחמדני ממקסם את הערכים.

אנחנו יודעים כעת איך להקצות מפרסמים למקומות.
נותר לנו להחליט איך לקבוע את התשלומים. עלינו למצוא דרך להכליל את מכרז ויקרי לכמה מוצרים.

הפתרון של גוגל: **מכרז מחיר שני מוכלל** GSP:
הגדרה:
– המפרסם שההכרזה שלו היא ה j בגובהה, זוכה במקום ה j, ומשלם את ההכרזה של המפרסם ה j+ 1.

כשיש מקום פרסום אחד הוא בדיוק כמו מכרז ויקרי. כשיש שני מקומות ויותר, האם הוא אמיתי ?

משפט: כשיש שני מקומות ויותר מכרז GSP אינו אמיתי.

אלגוריתם ויקרי – קלארק – גרובס VCG:
הנחות:
– יש מספר סופי של תוצאות אפשריות.
– לכל משתתף יש ערך כספי לכל תוצאה.
– התועלת היא ערך התוצאה פחות התשלום (קוואזי-לינארית).

האלגוריתם:
– בחר את התוצאה עם סכום הערכים הגבוהה ביותר.
– עבור כל שחקן:
  – חשב את סכום הערכים של שאר השחקנים.
  – חשב את סכום הערכים של שאר השחקנים אילו השחקן הנוכחי לא היה משתתף.
– גבה מהשחקן את ההפרש בין שני הסכומים.

במכרז עם חפץ אחד, VCG = מכרז מחיר שני:
– בחר את התוצאה עם סכום הערכים הגבוהה ביותר. (תן את החפץ לשחקן עם הערך הגבוהה ביותר V_1).
– עבור כל שחקן:
– חשב את סכום הערכים של שאר השחקנים כשהשחקן הנוכחי משתתף. (לזוכה, סכום האחרים הוא 0 , לכל אחד אחר הסכום הוא V_1).
– חשב את סכום הערכים של שאר השחקנים אילו השחקן הנוכחי לא היה משתתף. (לזוכה, הסכום של האחרים היה V_2 לכל אחד אחר V_1).
– גבה מהשחקן את ההפרש בין הסכומים. (לזוכה ההפרש הוא V_2, לכל אחד אחר 0).

משפט: אלגוריתם VCG הוא אמיתי.

חסרונות של VCG:
– קשה יותר למימוש.
– פחות ברור למפרסמים.
– עלול לגרות לירידה זמנית ברווחים.

יתרונות של VCG:
– יכול לטפל גם בדפים מורכבים, עם פרסומות בגדלים שונים, צורות שונות ומיקומים שונים.
– יכול לטפל גם במכרזים מורכבים, על קליק/לייק וכדומה.

לכן משתמשים בו בפרסומות של פייסבוק, גוגל ובאתרים אחרים.

<u>**מטלה – מכרזים אמיתיים**</u>

שאלה 1: **וריאציות על מכרז ויקרי** (**הקורס של טים, שאלות** 27, 29)

א. ויקי המציאה מכרז חדש למכירת חפץ יחיד: מי שההכרזה שלו גבוהה ביותר זוכה בחפץ, ומשלם את המחיר **השלישי** (במקום המחיר השני במכרז ויקרי). האם המכרז של ויקי אמיתי? הוכיחו.

ב. כדי לגייס כסף לשיפוץ הבית, החלטתם למכור את אוסף העטים הנדירים שלכם. באוסף יש 10 עטים זהים. ישנם 20 אנשים המעוניינים להשתתף במכרז. כל אחד מעוניין בעט אחד לכל היותר. תארו מכרז אמיתי ויעיל פארטו למכירת העטים.

תשובה 1:

א. המכרז הזה אינו אמיתי. דוגמה נגדית:

ההצעות: רמי–10 דני–5 צומי–3

רמי זוכה במוצר והתועלת שלו היא 3–10 =7, ולכל שאר המשתתפים התועלת היא 0.

במקרה זה, דני יכל להציע 11, לזכות במוצר והתועלת שלו תהיה 3–5 =2 כלומר אם הוא לא היה אומר את הערך האמיתי שלו, הוא היה מגדיל את התועלת שלו ולכן המכרז אינו אמיתי.

ב. נתשתמש במכרז מעטפות חתומות, כל משתתף יגיש הצעה במעטפה. מנהל המכרז יפתח את כל המעטפות ויסדר אותן בסדר יורד. עשרת ההצעות הגבוהות ביותר יזכו בעשרת העטים. הזוכרים ישלמו את הערך של ההצעה ה 11 בגודלה. המכרז הוא אמיתי, הוחכה מאוד דומה למכרז מחיר שני.

המכרז הוא יעיל פארטו מכיוון שהוא ממקסם את סכום הערכים.

שאלה 2: **מכרזי פירסום – איכות המודעה** (**הקורס של טים, תרגילים** 32, 34)

בשיעור הנחנו, שהסתברות ההקלקה על מודעה מסויימת תלויה רק במיקום שלה ולא במודעה.

במציאות, הסתברות ההקלקה על מודעה תלויה גם במיקום וגם במודעה עצמה.

א. נניח שלכל מודעה j יש מקדם איכות q_j, והסתברות ההקלקה כששמים אותה במקום k היא:

$r_{j,k} = r_k * q_j$

תארו אלגוריתם המשבץ מודעות למיקומים וממקסם את סכום הערכים. הוכיחו את נכונות האלגוריתם.

ב. נניח שלכל מודעה j ומיקום k יש הסתברות הקלקה כללית (לא דווקא ליניארית): $r_{j,k}$

הראו דוגמה שבה האלגוריתם מסעיף א לא עובד (לא מוצא שיבוץ הממקסם את סכום הערכים).

תשובה 2:

ממש דומה לאלגוריתם החמדני שהראנו בכיתה בתוספת מקדם האיכות:

לכל מפרסם j יש ערך הקלקה v_j וערך מקדם איכות q_j.

מכאן כל מפרסם מעריך את משבצת k כ: $v_j * r_k * q_j$

נשתמש באלגוריתם חמדני: סדר את המפרסמים בסדר יורד של v_j:

$v_1 * q_1 > v_2 * q_2 > \ldots$

תן למפרסם ה-j את המקום ה-j.

שאלה 3: **חשיפת מידע**

מצאתם בעליית-הגג שלכם שרשרת אבני-חן ישנה, ואתם מאד רוצים לדעת מה השווי שלה. השכן שלכם הוא מומחה לאבני-חן ויודע בדיוק מהו שווי השרשרת, אבל הוא לא רוצה להגיד לכם. הציעו מנגנון שיגרום לשכן שלכם לגלות לכם, מרצונו החופשי, את שוויה האמיתי של השרשרת. *רמז*: קיראו כאן: https://www.econ2.uni-bonn.de/pdf/papers/goethes_second.pdf

תשובה 3:
כדי לגלות את מחירה האמיתי של השרשרת מהשכן עלינו להציע לו להשתתף במכרז אמיתי כלשהו, מכיוון שהמכרז הוא מכרז אמיתי והשכן שיודע מהו השווי המדוייק של השרשרת הוא יציע במכרז את השווי האמיתי של השרשרת וכך נגלה זאת.

שאלה 4: **הקצאה אמיתית של מעונות סטודנטים**
משרד המעונות של האוניברסיטה מקצה מעונות לסטודנטים בעזרת האלגוריתם הבא:
כל סטודנט רושם את שלושת סוגי המעונות שהוא הכי רוצה, לפי הסדר.
המשרד מסדר את הסטודנטים לפי סדר עדיפות כלשהו (ותק, ציונים וכד').
המשרד עובר על הסטודנטים לפי הסדר, ונותן לכל סטודנט את החדר הכי גבוה בדירוג שלו שעדיין זמין. אם כל השלושה לא זמינים, הסטודנט מקבל חדר זמין כלשהו באקראי.
א. הוכיחו שהמנגנון אינו אמיתי.
ב. הוכיחו, שגם אם כל הסטודנטים מדווחים את ההעדפות האמיתיות שלהם, המנגנון אינו יעיל פארטו.
ג. הציעו שיפור קטן למנגנון, שיהפוך אותו לאמיתי וגם יעיל פארטו.

תשובה 4:
א. המנגנון אינו אמיתי, נניח ולאחד הסטודנטים יש מידע מוקדם על כך שרוב הסטודנטים מעדיפים חדר מסויים, נניח כי הסטודנט לא נמצא בראש סדר עדיפויות של המשרד. אזי, גם אם אותו הסטודנט רוצה את החדר המסויים הוא יעדיף להגיד את החדר השני שהוא הכי רוצה וכך להגדיל את הסיכויים שלו לקבל חדר יותר טוב. (כדאי לרגל אחר אחרים ולגלות אחר ההעדפותיהם).

ב. נניח כי כל הסטודנטים מדווחים על ההעדפות האמיתיות שלהם, נראה כי המנגנון אינו יעיל באמצעות דוגמה נגדית:

חדרים	חדר א'	חדר ב'	חדר ג'	חדר ד'	חדר ה'	חדר ו'
יוסי	20	19	18	17	16	15

במעונות יש 6 סוגי חדרים, יוסי שלח למשרד את העדפותיו חדר א'-חדר ב'-חדר ג', כשדגמו את יוסי חדרים א'-ג' היו כבר תפוסים. לכן יוסי קיבל באופן אקראי את חדר ו'. התוצאה לא יעילה פארטו בהנחה שיש מקום בחרדים ד',ה' שכל אחד מהם מהווה עבורו שיפור פארטו.

ג. נניח שיש N סוגים של חדרים במעונות, על המשרד לבקש מכל סטודנט להכליל בדירוג שלו את כל N הסוגים השונים של החדרים. המכרז הזה הוא אמיתי מכיוון שכל סטודנט מעדיף לרשום את הערך האמיתי שלו לגבי כל חדר, ולא לנסות להתכקם. כל שחקן שמגיע תורו מעדיף לקבל את החדר הכי טוב שזמין כרגע וזה בדיוק מה שאלגוריתם עושה במידה ואומרים את האמת.

שאלה 5: **הקצאת אמיתית של זמן חישוב**
כמה אנשים מעוניינים להריץ תהליכים על מחשב-על. לכל אדם יש תהליך אחד. לכל תהליך יש זמן ריצה שונה. מחשב-העל יכול להריץ תהליך אחד בכל פעם. המחשב פעיל במשך זמן מסויים (נניח 10 שעות) אבל זמן-הריצה הכולל של כל התהליכים הוא גדול יותר, כך שלא כל התהליכים יוכלו לרוץ. מנהל המערכת רוצה לתזמן את התהליכים (באופן סדרתי) כך שמספר התהליכים הכולל יהיה הגדול ביותר.
כל אדם יודע מה זמן-הריצה של התהליך שלו (כמה דקות הוא צריך), אבל מנהל-המערכת אינו יודע את זמני הריצה של התהליכים. עיזרו למנהל-המערכת לפתח אלגוריתם **אמיתי**, שיגרום לכל

אדם להצהיר על הזמן האמיתי של התהליך שלו, ויריץ את המספר הגדול ביותר של תהליכים במסגרת הזמן הקצוב.

להרחבה (לא חובה לצורך פתרון המטלה):

U Feige, M Tennenholtz (2011): "Mechanism design with uncertain inputs (to err is human, to forgive divine)" (2014): "On fair division of a homogeneous good"

תשובה 5:

כל אחד מדווח על הזמן שהתהליך שלו לוקח, האלגוריתם מסדר את התהליכים מהקצר ביותר לארוך ביותר ומתחיל להריץ אותם לפי הסדר. תהליך שזמנו נגמר אך התהליך עצמו עדיין רץ, קוטעים אותו ולא מגלים את המידע שהושג עד כה בחישוב. ממשיכים לתהליך הבא, עד שהזמן נגמר. כך אף אחד לא ירצה להגיד זמן גבוהה יותר ממה שהוא צריך כי זה רק ידרג אותו במקום פחות טוב. וגם אף אחד לא ירצה להגיד זמן קצר יותר מהזמן האמיתי שהתהליך שלו לוקח מכיוון שאז התהליך יקטע ולא ישיג שום מידע. ולכן כל אחד יצרה להגיד את הזמן האמיתי שיקח לתהליך שלו.

שאלה 6: **עיצוב דף פירסום**

בדף-אינטרנט מסויים אפשר לשים **או** פירסומת אחת ארוכה, **או** שתי פרסומות קצרות.

ישנם מספר מפרסמים המתחרים על מקום בדף. ההעדפות של כל מפרסם מיוצגות ע"י המחלקה:

```
class Advertiser {
	float  longvalue;
	// כמה המפרסם מרויח (בשקלים) אם פרסומת ארוכה שלו מופיעה בדף
	float shortvalue;
	// כמה המפרסם מרויח (בשקלים) אם פרסומת קצרה שלו מופיעה בדף
};
```

מפרסם שאינו מופיע בדף מרויח 0 שקלים. מנהלי האתר רוצים שהמפרסמים יהיו מרוצים – המטרה שלהם היא למקסם את סכום רווחי המפרסמים. עיזרו להם להחליט איזה פירסומות לשים באתר!

א. כיתבו אלגוריתם, בעברית או בפסאודו-קוד, המקבל כקלט את רשימת המפרסמים, ומחזיר כפלט:

כמה פירסומות יהיו בעמוד (אחת או שתיים);

איזה מפרסם/מפרסמים יופיעו בעמוד.

ב. תכננו מכרז הממקסם את סכום רווחי המפרסמים, וגם מעודד כל מפרסם לחשוף את הערכים longvalue, shortvalue **האמיתיים** שלו. המנגנון מקבל כקלט את רשימת המפרסמים, ומחזיר כפלט: כמה פירסומות יהיו בעמוד, איזה מפרסם/מפרסמים יופיעו בעמוד, ואיזה מחיר ישלם/ישלמו המפרסם/מפרסמים למנהל האתר (בשקלים).

הערה: אם מפרסם מסויים מופיע בדף ומשלם מחיר, התועלת שלו היא הרווח מהפירסום פחות המחיר.

ג. הדגימו את פעולת המכרז שכתבתם על הקלט הבא, ובו ארבעה מפרסמים:

```
ad[0].longvalue= 10;   ad[0].shortvalue= 8;
ad[1].longvalue= 9;    ad[1].shortvalue= 1;
ad[2].longvalue= 8;    ad[2].shortvalue= 4;
ad[3].longvalue= 7;    ad[3].shortvalue= 3;
```

תשובה 6:
א. קלט: רשימת מפרסמים.
פלט: כמה פיסומות יהיו בעמוד, איזה מפרסם/ מפרסמים יופיעו בעמוד.
האלגוריתם:
1. נגדיר שני מערכי עזר אחד שיכיל את כל התוצאות של המפרסמים עבור פרסומת קצרה, והשני עבור פרסומת ארוכה.
2. נעבור על כל המפרסמים ונמלא את הערכים במערכי העזר.
3. נמצא את המקסימום במערך העזר של הפרסומות הארוכות ואת האינדקס שלו.
4. נמצא את המקסימום הראשון והשני במערך העזר של הפרסומות הקצרות ואת האינדקסים שלהם.
5. נבדוק מה יותר גדול, ערך הפרסומות הארוכה הכי גדולה, או ערך שני הפרסומות הקצרות הכי גדולות.
6. נחזיר את התוצאות. (האינדקס = המפרסם)

ב. נשתמש במכרז VCG, מכיוון שלמפרסמים שני פרמטרים, ואנו רוצים למקסם את סכום ערכיהם. המנגנון תחילה ימצא באמצעות האלגוריתם שבסעיף א את המפרסם/ים. עבור כל מפרסם נחשב את סכום הרווחים של המפרסמים האחרים איתו ובלעדיו, עבור כל שחקן נגבה את ההפרש הזה.

ג.
עבור הקלטים הנתונים, האלגוריתם היה בוחר להציג שתי פרסומות קצרות של מפרסם 0 ו 2.
התשלום שהיה נגבה מהם הוא:
עבור 0- הרווח של האחרים הוא 4. בלי 0 היינו בוחרים את מפרסם 1 פרסומת ארוכה, שערכה הוא 9.
לכן 0 משלם 4-9 =5
עבור 2- הרווח של האחרים הוא 8. בלי 2 , היו נבחרות שתי פרסומות קצרות שערכם הוא 11 .
ולכן 2 משלם 8-11 =3.

<u>מכרזים אלגוריתמיים (שיעור 6):</u>

מציאת מסלול זול ביותר:
נתונה רשת. לכל קשת יש עלות מעבר. צריך להעביר חבילה בין שתי נקודות ברשת, במסלול הזול ביותר.
אם עלות כל קשת ידועה לכולם- נשתמש באלגוריתם (דייקסטרה).
אם עלות כל קשת ידועה רק לבעליה- נשתמש במכרז (VCG).

בעיית התרמיל:
מכניסים אותכם לחדר מלא בחפצים, נותנים לכם תרמיל שיכול להכיל עד 100 ק"ג, ואומרים לכם "כל מה שתצליחו להכניס לתרמיל שלכם".
לכל חפץ יש משקל וערך שונה.
- אם הערך של כל חפץ ידוע לכולם, אלגוריתם.
- אם הערך של כל חפץ ידוע רק לחפץ, מכרז.

מכרז VCG **לבעיית התרמיל:**
כשיש M חפצים, צריך לפתור M+ 1 בעיות תרמיל.
הבעיה: בעיית התרמיל היא בעייה NP-קשה!
פתרון אפשרי: אלגוריתמי קירוב.

אלגוריתם חמדני א':
- סדר את החפצים בסדר יורד של הערך.
- בחר חפצים לפי הסדר עד שהתרמיל מתמלא.

דוגמה נגדית:

100$/100K, 20$/2K, 20$/2K, 20$/2K, … …

הראשון יזכה וישלם 1000$ שזה יותר מהערך שלו!

אלגוריתם חמדני ב':
- סדר את החפצים בסדר היורד של ערך חלקי משקל.
- בחר חפצים לפי הסדר עד שהתמיל מתמלא.

דוגמה נגדית:

**20$/2K,** 100$/100K

אלגוריתם א' + ב': הפעל את שני האלגוריתמים החמדניים. בחר את התוצאה עם הסכום הגבוהה.

משפט: אלגוריתם א' + ב' נותן קירוב של ½.

מכרז מיירסון (Myerson):
נתונים:
- ישנו כלל בחירה הקובע האם השחקן נבחר או לא.
- לכל משתתף יש ערך ל"היבחרות".

דרוש: כלל תשלום, שאיתו המכרז יהיה אמיתי.
שאלה: האם לכל כלל בחירה קיים כלל תשלום אמיתי ?

הגדרה: כלל בחירה נקרא מונוטוני אם, עבור כל שחקן j, הכלל הוא פונקציה מונוטונית עולה של v_j.
כלל בחירה בינארי הוא מונוטוני אם עבור כל שחקן j, אם הוא נבחר כשהערך שהוא הוא X, אז הוא נבחר גם כשהערך שלו הוא כל מספר הגדול מ-X.

דוגמאות לכללים מונוטוניים:
- בחר את שלושת הערכים הגדולים ביותר.
- בחר את הערך הגדול ביותר, בתנאי שהוא מעל 10.

משפט מיירסון: מונוטוניות היא תנאי הכרחי ומספיק לאמיתות.
כלומר: אם כלל בחירה הוא לא מונוטוני, אין כלל תשלום אמיתי. ואם כלל בחירה הוא מונוטוני אזי יש כלל תשלום אמיתי והוא יחיד.

ערך הסף: הערך שבו פונקציית כלל הבחירה מתחלפת מ-0 ל-1.(מזכה ללא זכה).
לכן כלל התשלום יהיה ערך הסף.

מצאנו כלל תשלום אחד ויחיד המועמד להיות אמיתי:
- לכל שחקן j יש ערך סף מסויים t_j. הנקבע לפי כלל בחירה.
- אם $v_j > t_j$, אז השחקן נבחר ומשלם t_j.
- אחרת, השחקן לא נבחר ולא משלם.

הנחה: המשקל של כל משתתף ידוע. כל משתתף צריך רק להגיד את הערך שלו.

	וק"ג	מיירסון
פרמטרים לכל שחקן	הרבה *(למשל: בחירת מסעדה)*	אחד
כלל בחירה	מיקסום סכום ערכים	כל כלל מונוטוני *(למשל: קירוב בעיית התרמיל, מיקסום רווח)*

מטלה – מכרזים: וק"ג, פירסום, מיירסון

שאלה 1: שוק דו-צדדי
אחד השימושים של תורת המכרזים הוא ניהול אוטומטי של שוק דו-צדדי, כמו בבורסה. בבורסה יש הרבה קונים, הרבה מוכרים, והרבה חפצים (מניות). אבל כדי להבין מה קורה, אנחנו נתייחס לגירסה מוקטנת של השוק, שבה יש רק קונה אחד, מוכר אחד, וחפץ אחד.
מוכר אחד מגיע לשוק עם חפץ אחד. ערך החפץ בעיני המוכר הוא s.
קונה אחד מגיע לשוק, הוא רואה את החפץ וחושב שהוא שווה b.
אתם, מנהלי השוק, צריכים להחליט:

האם המוכר ייתן את החפץ לקונה, או יחזור עם החפץ הביתה?
כמה כסף ישלם הקונה?
כמה כסף יקבל המוכר?
כמה כסף יישאר לכם בכיס כ"דמי תיווך"?

א. הציעו מכרז אמיתי ויעיל-פארטו, המבוסס על VCG, לפתרון הבעיה. הסבירו בפירוט מה יעשה המכרז בכל אחד מהסעיפים 1-4.
ב. מה הבעיה במכרז שתיארתם בסעיף א?

תשובה 1:
א. נרצה למקסם את סכום הערכים. לכן נמכור את החפץ אם ורק אם b>s.
תשלום הקונה: הוא s
המוכר יקבל b.
דמי תיווך: s-b
ב. התוצאה במכרז VCG בשוק דו-צדדי היא עם גירעון עבור מהנל השוק.

שאלה 2: בעיית תחנת המוניות עם נהגים אסטרטגיים
בתחנת מוניות עובדים שלושה נהגים. כל נהג נמצא עכשיו במקום אחר. התחנה מקבלת בו-זמנית שלוש פניות מנוסעים הנמצאים במקומות שונים. התחנה צריכה להחליט איזה נהג לשלוח לאיזה נוסע. הנתונים:
המרחק בין נהג i לבין נוסע j הוא: d[i,j] (ק"מ).
התחנה לא יודעת איפה נמצא כל נהג, ולכן לא יודעת את d[i,j]. הערך הזה ידוע רק לנהג i.
מחיר של דלק לקילומטר הוא p (ש"ח).
א. תארו מכרז להתאמת נוסעים לנהגים, עם התכונות הבאות:
אמיתי – לכל נהג i כדאי לדווח את המרחק האמיתי בינו לבין כל נוסע (d[i,j] לכל j).
יעיל – המכרז מוצא את ההשמה הזולה ביותר של נהגים לנוסעים.
ב. הדגימו את פעולת המכרז: הסבירו מה מדווחים הנהגים, איזו השמה מחשב האלגוריתם, וכמה כל אחד משלם/מקבל.

שאלה 3: הפיכת אלגוריתם למכרז
מיצאו אלגוריתם נוסף שאתם מכירים, שונה מהאלגוריתמים שנלמדו בכיתה, שאפשר להפוך אותו למכרז. תארו בפירוט את הקלט למכרז, את הערכים של המשתתפים, את כלל-הבחירה ואת כלל-התשלום.

תשובה 3:
להפוך כל אלגוריתם הממקסם או מוצא מינימום של ערכים למכרז VCG. לדוגמה: מציאת עץ פורש מינימאלי (kruskal, prim) או מציאת מסלול מינימאלי/מקסימלי (פלוייד ורשאל/דייקסטרה).

שאלה 4: **תיכנות – כלל-התשלום של מיירסון**
נתונה פונקציה המייצגת כלל-בחירה. כותרת הפונקציה בפסאודו-Java היא:

```
bool[] choices (double[] values);
```

הפונקציה מקבלת כקלט וקטור של מספרים המייצגים את הערכים של השחקנים (בשקלים).
הפונקציה מחזירה כפלט וקטור בוליאני המחזיר, לגבי כל שחקן, האם הוא נבחר או לא.
כיתבו פונקציה המייצגת את כלל-התשלום המתאים, לפי משפט מיירסון. כותרת הפונקציה:

```
double[] payments (double[] values);
```

הפונקציה מקבלת כקלט את וקטור הערכים (כמו הפונקציה הקודמת), ומחזירה את וקטור התשלומים, בשקלים, ברמת-דיוק של אגורה אחת (0.01 ש"ח).
הערות:
אם הפונקציה מגלה שכלל-הבחירה אינו מונוטוני – היא צריכה לזרוק חריגה.
לא חייבים לכתוב בשפת Java תקנית – אפשר לכתוב בשפה אחרת או פסאודו-קוד, העיקר שיהיה ברור מה עשיתם.

תשובה 4:
כלל התשלומים המתאים הוא ערך הסף עבור הנבחרים, ו-0 עבור מי שלא נבחר.
האלגוריתם:
1. נשלח את וקטור המחירים לפונקציה choices ונשמור את התשובה במערך.
2. נגלה את ערך הסף של פונקציית הבחירה באמצעות שימוש באלגוריתם מעבר החצייה.
3. לאחר שגילינו את ערך הסף, עבור כל שחקן שנבחר ע"י הפונקציה choices ב 1. המחיר שישלם יהיה ערך הסף שלו, עבור כל שחקן שלא נבחר התשלום שלו יהיה 0.

שאלה 5: **תיכנות – מכרז מיירסון לבעיית התרמיל**
נתון מערך גלובלי:

```
double[] weights;
```

שבו האיבר ה-i מייצג את המשקל של השחקן ה-i.
א. בחרו אחד מאלגוריתמי-הקירוב שלמדנו (א, ב, או א +ב), וכיתבו את פונקציית-הבחירה המתאימה לו:

```
bool[] choices (double[] values);
```

ב. בחרו אחד מאלגוריתמי-הקירוב שלמדנו (א, ב, או א +ב), וכיתבו את פונקציית-התשלום המתאימה לו:

```
double[] payments (double[] values);
```

כאן (בניגוד לשאלה הקודמת) חישוב התשלומים צריך להיות מדוייק, ע"י נוסחה מפורשת.

תשובה 5:

א. אלגוריתם קירוב חמדני א:

```
bool[] choices(double[] values)
   sort(values) in decreasing order of values
   totalWeight = 0
   ans[] bool array
   for( I= 0 to values.length and totalWeight <=  bagWeight)
      if(values[i] +  totalWeight <=  bagWeight)
         totalWeight + =  values[i]
         ans[i] =  true
      end-if
   end-for
return ans
```

ב.

עבור כל איבר שנכנס לסל נמצא את ערך הסף שלו, ניתן על פה להשתמש בשיטת החצייה, לשלוח את מערך המשקלים כל פעם ל choices ולראות מתי כל ערך עובר מ true ל false, זהו ערך הסף שלו.(נמצא את ערך הסף בקירוב- לפי איזה קירוב שנרצה נוכל להגדיר זאת באלגוריתם החצייה)

מכרזים למיקסום רווח (שיעור 7):

מיקסום רווח לעומת מיקסום סכום ערכים:

נניח שאנו מוכרים חפץ אחד ויש קונה אחד.
שאלה: איזה מכרז ממקסם את סכום הערכים?
תשובה: נותנים לו את החפץ בחינם.
שאלה: איזה מכרז ממקסם את הרווח של המוכרים?
תשובה: זה קשה- תלוי בערך של השחקן!

במקום למקסם רווח- ננסה למקסם את תוחלת הרווח. זה דורש מידע סטטיסטי על ערכים של שחקנים ("סקר שוק").

דוגמה: סקר שוק הראה שערכי הקונים מתפלגים אחיד בין 10 ל-30. מהי תוחלת הרווח של מכרז מיירסון? - זה תלוי בכלל הבחירה:
- אם כלל הבחירה הוא "מקסם את סכום הערכים". אז ערך הסף הוא 0, התשלום הוא 0, תוחלת הרווח היא 0.
- אם כלל הבחירה הוא "בחר את הקונה אם ורק אם הערך שלא הוא לפחות 10". ערך הסף הוא 10, הקונה נבחר בהסתברות 1 (תמיד), תוחלת הרווח היא 10.
- אם כלל הבחירה הוא " בחר את הקונה אם ורק אם הערך שלו הוא לפחות 15". ערך הסף הוא 15, הקונה נבחר בהסתברות ¾, תוחלת הרווח היא ¾ * 15 =11.25.
- באופן כללי, ניתן למצוא את כלל הבחירה כך: "בחר את הקונה אם ורק אם הערך שלו לפחות p". ערך הסף הוא p, הקונה נבחר בהסתברות:

$$(30-p)/(30-10)$$

ותוחלת הרווח היא:

$$p * (30-p)/(30-10) = (30p-p^2)/20$$

אנו רוצים למצוא את המקסימום ולכן נגזור לפי p:

$$(30-2p)/20 = 0 \rightarrow 2p=30 \rightarrow p=15$$

הכללה: נניח שאספנו נתונים סטטיסטיים על ערכי הקונים, והגענו למסקנה שהם מתפלגים כ:

$$Prob[v<p] = F(p)$$

איזה מחיר ממקסם את תוחלת הרווח?

$$E[p] = p*Prob[v>p] = p*(1-F(p))$$

אנו רוצים למצוא את המקסימום ולכן נגזור לפי p ונשווה ל-0:

$$p-(1-F(p))/F'(p) = 0$$

נגדיר את פונקציית הערך הוירטואלי:

$$r(v) := v-(1-F(v))/F'(v)$$

המכרז האופטימלי הוא: מכור את החפץ אם ורק אם $r(v)>0$.

מקסום רווח בשיטת מיירסון- כללי:

נתון: שוק חד פרמטרי:
- לכל משתתף יש ערך כספי יחיד ל"היבחרות".
- הערך של משתתף j לקוח מההתפלגות F_j.

דרושים:
- כלל בחירה לבחירת תת קבוצה של משתתפים.
- כלל תשלומים שאיתו כלל הבחירה הוא אמיתי.

כזכור לפי משפט מיירסון, ברגע שיש כלל בחירה, יש כלל תשלומים (מחיר הסף) יחיד שאיתו הכלל אמיתי.
נחפש כלל בחירה הממקסם את תוחלת הרווח.
תוחלת הרווח היא סכום הערכים הוירטואליים.
מסקנה: כדי למקסם רווח, צריך למצוא כלל בחירה הממקסם את סכום הערכים הוירטואליים.

מיקסום רווח בשיטת מיירסון –חפץ אחד:
תוחלת הרווח היא סכום הערכים הוירטואליים.

$$r_j(v) := v-(1-F_j(v))/F_j'(v)$$

א. קונה אחד:
תוחלת הרווח = הערך הוירטואלי = $r(v)$
כלל הבחירה הוא: מכור אם ורק אם $r(v)>0$
התשלום הוא ערך הסף $r(v)=0$

ב. הרבה קונים עם אותה התפלגות F ועם אותו r:
תוחלת הרווח: היא $r(v_j)$ של המנצח.
כלל הבחירה: הוא מכור למשתתף עם v_j הכי גבוהה בתנאי ש $r(v_j)>0$
כלל התשלום הוא: הערך השני בגובהו או $r(v)=0$ – הגבוהה מביניהם.

ג. שני קונים עם התפלגויות שונות:
תוחלת הרווח = $r_l(v_j)$ של המנצח.
כלל הבחירה: הוא מכור למשתתף עם $r_l(v_j)$ הכי גבוהה, בנתאי ש $r_l(v_j)>0$.
התשלום = ערך הסף.
דוגמה: $F_a=\text{Unif}[10, 30]$, $F_b = \text{Unif}[20, 40]$

$$r_a(v) = 2v-30 \ , \ r_b(v) = 2v-40$$

אם a אומר 23 ו– b אומר 27, אז a יזכה! וישלם את ערך הסף שלו שהוא 22.

<u>מטלה – מכרזים למיקסום רווח</u>

שאלה 1: מכרז מסובסד
אתם קבלני-בניין ובזה הרגע סיימתם לבנות דירה. אתם מעוניינים למכור אותה באופן שיעשה אתכם כמה שיותר עשירים (בתוחלת).
יש הרבה קונים פוטנציאליים, וכל קונה מייחס לדירה ערך שונה. מסקר-שוק שביצעתם, התברר שהתפלגות הערכים באוכלוסיה היא התפלגות אחידה בין 0 ל-1000 [באלפי ש"ח].
קונים צעירים במיוחד או מבוגרים במיוחד מזכים את הקבלן בהשתתפות ממשרד השיכון באופן הבא:
קבלן המוכר דירה לקונה בן 20 ומטה – מקבל ממשרד השיכון 100 אלף ש"ח.
קבלן המוכר דירה לקונה בן 60 ומעלה – מקבל ממשרד השיכון 150 אלף ש"ח.
תארו מכרז אמיתי שימקסם את תוחלת הרווח שלכם. כיתבו את המנגנון בשפת-תיכנות לבחירתכם או בפסאודו-קוד. הניחו שקיימת המחלקה הבאה המייצגת קונה פוטנציאלי; ניתן להוסיף לה שיטות לפי הצורך.

```
class Buyer {
        int age();      // גיל בשנים
        int value();   // ערך באלפי ש"ח
};
```

כותרת הפונקציה המבצעת את המנגנון:

```
void sellHouse(Buyer[] buyers) { ... }
```

תשובה 1:
נשתמש במנגנון מיירסון, נבדוק מהו ערך הסף שיקנה לי תוחלת רווח מקסימלית:
עבור קונה בן 20 ומטה נמכור לו אם"ם הערך שלו מעל 400.
עבור קונה בן 60 ומעלה נמכור לו אם"ם הערך שלו מעל 350.
אחרת נמכור לו אם"ם הערך שלו הוא מעל 500.

```
class Buyer{
        int age();
        int value();
        int bonus(){
                if(age() <=  20) return 100
                else if(age() >=  60) return 150
                else return 0
        }
        int virtual_value(){ return 2*value() – 1000 }
        int profit(){ return virtual_value() +  bonus() }
}
```

```
void sellHouse(Buyer[] buyers){
        sort(buyers) in decreasing order of profit()
        if(buyers[0].profit() < 0)
                print("The house was not sold!")
        else{
                print(The house sold to buyers[0])
                if(buyers[1].profit() > 0)
                        return ( buyers[1].profit() +  1000 — buyers[1].bonus() )/2
                else{ return  (1000 – buyers[1].bonus() )/2
        }
}
```

שאלה 2: **מיקסום רווח עם ברירת-מחדל**

מצאתם ברחוב ציור עתיק. בחנות יד שניה הציעו לכם עבורו X ש"ח. אתם רוצים להשיג סכום גבוה יותר ע"י מכירה לאספן עתיקות ידוע, שהערך שלו לציור מתפלג לפי פונקציה F.

תארו מנגנון אמיתי הממקסם את הרווח שלכם ממכירת הציור. שימו לב – המנגנון תלוי ב-F וגם ב-X.

תשובה 2:

הערך הוירטואלי של האספן: $r(v) := v-(1-F(v))/F'(v)$

אם נמכור לאספן תוחלת הרווה שלנו תהיה $r(v)$

אם נמכור לחנות תוחלת הרווח שלנו תהיה x

לכן נמכור לאספן אם ורק אם $r(v)>x$

והאספן יצטרך לשלם את מחיר הסף המחיר בו $r(v)=x$.

שאלה 3: **מכרז לקנייה**

אתם מנהלים את מחלקת הרכש ברכבת ישראל. קיבלתם הוראה לקנות קרון חדש במחיר נמוך ככל האפשר. יש כמה חברות המייצרות קרונות, לכל חברה יש עלות אחרת לייצור קרון. אתם לא יודעים את העלויות של החברות השונות, אבל מתוך נתונים סטטיסטיים שאספתם, אתם יודעים שעלות-הייצור מתפלגת לפי פונקציה F (התפלגות זהה עבור כל החברות).

הנהלת הרכבת מעריכה, שהתועלת שתפיק מהקרון היא U (מספר ידוע – נניח מיליארד ש"ח).

תארו מנגנון אמיתי לקניית קרון, שבו תוחלת התועלת של רכבת-ישראל תהיה מקסימלית.

שאלה 4: **ערך וירטואלי בהתפלגות אחידה**

נניח שהערך של קונה מסויים מתפלג אחיד בין a ל- b (שני פרמטרים חיוביים).

א. כיתבו ביטוי לפונקציית הערך הוירטואלי של הקונה, $r(v)$, כפונקציה של a, b.

ב. כיתבו ביטוי למחיר האופטימלי למכירת חפץ כלשהו לקונה זה, $r^{-1}(0)$.

ג. כיתבו ביטוי לתוחלת הרווח של המוכר כאשר הוא משתמש במחיר האופטימלי.

תשובה 4:

א. $r(v)= 2v -b$

ב. נמכור לקונה אם"ם $v>= b/2$

ג. אם $a>b/2$, אז המחיר תמיד a, הקונה קונה בהסתברות 1 וישלם a, תוחלת הרווח היא a.

אחרת, הקונה קונה בהסתברות:

$(b-b/2)/(b-a) = (b/2)/(b-a) = b/2(b-a)$

ומשלם b/2 לכן תוחלת הרווח היא:

$[b/2(b-a)] * b/2 = b^2/4(b-a)$

שאלה 5: התפלגות אמפירית וערך וירטואלי

כפי שלמדנו בכיתה, מכרז מיירסון למיקסום רווח משתמש בפונקציית הערך הוירטואלי, והיא משתמשת בפונקציית התפלגות ההסתברות:

$F(x) = Prob[v < x]$

$r(x) = x - [1-F(x)]/F'(x)$

ברוב המקרים, הפונקציה F אינה ידועה, ואנחנו צריכים לחשב אותה בקירוב מתוך נתונים סטטיסטיים.

כיתבו מחלקה לחישוב פונקציה זו. במחלקה יהיו לפחות שלוש שיטות:

איתחול (בנאי) – מקבל וקטור של ערכים (שנאספו בסקרי-שוק).

F – מקבלת ערך x, ומחזירה את ההסתברות האמפירית שהערך יהיה קטן מ-x. *שימו לב* – הפונקציה F תמיד מחזירה ערך בין 0 ל-1.

r – מקבלת ערך x, ומחזירה את הערך הוירטואלי המתאים.

הוסיפו שיטות נוספות לפי הצורך.

```
class Distribution {
	Distribution(int[] values);
	double F(int x);
	double r(int x);
}
```

חלוקת עלויות (שיעור 8):

דוגמה: שיתוף נסיעה במונית.

נסיעה משותפת במונית יכולה לחסוך בעליות.

- אם כל הנוסעים עושים את אותו המסלול, הגיוני לחלק את דמי הנסיעה שווה בשווה.
- אבל מה עם כל אחד מהנוסעים נוסע במסלול אחר?

שאלה: איך לחלק את דמי הנסיעה בין הנוסעים? (חלוקה הוגנת)

שאלה: איך להחליט מי ישתתף בנסיעה? (מכרז)

בעיה כללית: משחק שיתופי:

נתונים:

- קבוצה של שחקנים N.
- לכל תת קבוצה S – העלות של מתן שירות רק לקבוצה הזאת היא C(S).

המטרה: לגבות מכל שחקן j תשלום p(j), כך שסכום התשלומים הכללי הוא C(N). כלומר, התשלומים מכסים את העלות של כל הקבוצה.

עלות שולית: העלות השולית של שחקן j, ביחס לקבוצת שחקנים S, היא התוספת שהוא מוסיף לעלות כשהוא מצטרף לקבוצה:

$$c= (S \cup \{j\}) - c(S)$$

עקרון ההגינות: כלל תשלום נקרא סימטרי אם הוא תלוי רק בעלויות השוליות. אם לשני שחקנים יש עלויות שוליות זהות ביחס לכל הקבוצות, אז הם צריכים לשלם אותו מחיר.

עקרון הלינאריות: אם מכפילים את העלויות בקבוע – כל התשלומים נכפלים באותו קבוע. אם מחברים שתי טבלאות עלויות – כל התשלומים מתחברים.

עקרון העציץ: שחקן שכל העלויות השוליות שלו הן 0, משלם 0.

משפט שאפלי:

משפט: ישנו כלל תשלומים אחד ויחיד המקיים את כל שלושת העקרונות: עקרון ההגינות, העציץ והלינאריות. כלל התשלומים הזה נקרא ערך שאפלי.

אלגוריתם לחישוב ערך שאפלי:

- לכל אחד מ N! הסידורים האפשריים:
- עבור כל שחקן:
  - חשב את העלות השולית שלו בסידור הזה.
- חשב את הממוצע של N! העלויות השוליות, זהו הערך שאפלי.

מצאנו דרך לחלק את העלויות בין הנוסעים.

איך מחליטים מי יקבל שירות?

- עד כה הנחנו שכולם נוסעים.
- אבל מה קורה אם העלות גבוהה מדי עבור חלק מהנוסעים? איך מחליט מי יסע?

נתון:
– לכל שחקן j, ערך הנסיעה הוא v_j.
– אם תת קבוצה מסויימת נוסעת, הרווחה החברתית היא סכום הערכים של הנוסעים בתת הקבוצה, פחות עלות הנסיעה.

דרוש:
– כלל החלטה שהוא יעיל פארטו: ממקסם את הרווחה החברתית.
– כלל החלטה שהוא אמיתי: מעודד את כל שחקן j, לגלות את v_j שלו.

מכרז לקבלת שירות בשיטת VCG:
– התוצאות האפשריות, כל 2^n תת הקבוצות האפשריות.
– הערך של שחקן j הוא v_j אם נוסע, אחרת 0.
– הערך של הנהג הוא מינוס עלות הנסיעה.
– בוחרים את התוצאה הממקסמת את הסכום.
– תשלום שחקן j: = הסכום בלי j, פחות הסכום של האחרים כש j נמצא.

משפט: מכרז VCG הוא: יעיל פארטו, אמיתי, הבעיה היא שהוא יכול ליצור גירעון!

מכרז לקבלת שירות מולין שנקר:
קלט: כלל תשלום, p(S,j) = כמה משלם שחקן j אם הקבוצה שמקבלת שירות היא S.
אלגוריתם:
1. איתחול: כולם נכנסים לחדר.
2. אומרים לכל אחד כמה הוא צריך לשלם לפי p, בהנחה שכל הנדוכחים בחדר משתתפים.
3. מי שחושב שזה יקר מידי – יוצא מהחדר.
4. אם מישהו יצא מהחדר – חזור לצעד 2.
5. אחרת – סיים ושלח את הנשארים למונית.

האם מכרז מולין שנקר הוא אמיתי?
הגדרה: כלל תשלום p נקרא מונוטוני אם כשהקבוצה קטנה, התשלום גדל או נשאר זהה.
If S <= T, then for all j: p(S,j) >= P(T,j)

משפט: ההדמייה של מכרז מולין שנקר עם כלל תשלום מונוטוני היא אמיתית.

הגדרה: פונקציית עלות נקראת תת מודולארית אם יש לה עלות שולית פוחתת, כלומר:
If S <= T, then c(S U {j}) – c(S) >= c(T U {j}) – c(T)

משפט: במשחק עם עלות שולית פוחתת, כלל התשלום של שאפלי הוא מונוטוני.

יתרונות:
– מאוזן תקציבית.
– הוגן.
– אמיתי (אם יש עלות שולית פוחתת).

חסרון:
– לא יעיל פארטו.

	יעיל פארטו	מאוזן תקציבית	אמיתי
VCG	כן	לא	כן
מולין-שנקר	לא	כן	כן
תשלום = ערך	כן	כן	לא

<u>מטלה – חלוקת עלויות ומכרזי שירות</u>

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.

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?

תשובה 2:
א. נחשב את העלות עבור כל אחד מהמשתתפים לנסיעה מהבית לאריאל:
יערה, כפר תבור:532
שי, כפר סבא:183
אתי, אריאל: 32.
ב. המסלול הקצר היותר יהיה יערה-שי-אתי ומחירו 635.
ג. חישוב עלות כל נוסע לפי ערך שאפלי:

סדר:	יערה-שי-אתי	יערה-אתי-שי	שי-יערה-אתי	שי-אתי-יערה	אתי-יערה-שי	אתי-שי-יערה	ממוצע
יערה	532	532	420	423	529	423	476.5
שי	71	74	183	183	74	180	127.5
אתי	32	29	32	29	32	32	31
סכום	635	635	635	635	635	635	635

ג. שימוש הערך שאפלי משתלם עבור כל אחד מהשחקנים כי כל אחד משלם פחות ממה שהוא היה צריך לשלם אם היה נוסע לבד.

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 בעשיריה הראשונה.

תשובה 3:
נחשב לפי ערך שאפלי:

סדר:	א-ב-ג	א-ג-ב	ב-א-ג	ב-ג-א	ג-א-ב	ג-ב-א	ממוצע
א	0	0	4	5	6	5	20/6
ב	4	4	0	0	4	5	17/6
ג	6	6	6	5	0	0	23/6
סכום	10	10	10	10	10	10	10

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*:

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

הגדרה: פונקציה v על קבוצות נקראת *תת-חיבורית* (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

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 בונוס 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.

אלגוריתמי שידוך (שיעור 9):

שאלה: שיבוצים ושידוכים, מה ההבדל בין הבעיות ?
- שיבוץ סטודנטים לחדרים בדירה שכורה.
- שידוך סטודנטים למחלקות באוניברסיטה.

תשובה:
- הבעיה הראשונה היא חד צדדית, רק לסטודנטים יש העדפות (לחדר לא אכפת מי יגור בו).
- הבעיה השניה היא דו-צדדית, גם למחלקות יש העדפות.

שוק דו-צדדי:
שוק דו-צדדי הוא שוק שבו צריך להתאים בין המשתתפים משתי קבוצות, כאשר לכל משתתף מכל אחת מהקבוצות יש העדפות שונות.
בשוק דו-צדדי אי אפשר לצפות להגינות.
נחפש את התכונות הבאות: יעילות פארטו, יציבות.

שידוכים יציבים:
זוג מערער: סטודנט ומחלקה שאינם משודכים, והם מעדיפים זה את זו על פני השידוכים הנוכחיים שלהם.
שידוך יציב: שידוך בלי זוגות מערערים.
הגדרת הבעיה:
- מספר הסטודנטים והמחלקות שווה.
- בכל מחלקה יש מקום לסטודנט אחד.
- לכל מחלקה יש סדר העדפות על הסטודנטים.
- לכל סטודנט יש סדר העדפות על המחלקות.
- האם תמיד קיים שידוך יציב ? כן, ויש אלגוריתם יעיל שמוצא אחד כזה.

אלגוריתם "קבלה על תנאי":
א. כל סטודנט הולך למחלקה שהוא הכי רוצה, מבין המחלקות שעדיין לא דחו אותו.
ב. כל מחלקה "מקבלת על תנאי" את הסטודנט שהיא הכי רוצה, מבין אלה שנמצאים בה, ודוחה את כל השאר.
ג. חוזרים על שלבים א ו-ב עד אשר כולם משודכים.

משפט: האלגוריתם מסתיים בשידוך יציב.

מי מציע ומי מקבל ? באלגוריתם שהצגנו למעלה הסטודנטים מציעים למחלקות, והמחלקות מחליטות את מי לקבל. יכלנו גם לנהל את ההלגוריתם בסדר הפוך: המחלקות מציעות לסטודנטים, והסטודנטים מחליטים את מי לקבל.

משפט: כשסטודנטים מציעים, השידוך המתקבל הוא הטוב ביותר לכל הסטודנטים מכל השידוכים היציבים.

אלגוריתם "קבלה על תנאי" במחלקות גדולות:

מה עושים כשבכל מחלקה יש כמה מקומות ?

– כל מחלקה "מקבלת על תנאי את הסטודנטים הטובים ביותר שפנו אליה, עד לכמות המקומות הפנויים אצלה, ודוחה את כל השאר.

זה עובד כשאין תלות בין הסטודנטים. זה לא עובד למשל כאשר:

– זוגות סטודנטים נשואים הרוצים להתקבל לאותה המחלקה.
– מחלקות רוצות לקבל מספר מסויים של סטודנטים מקבוצות מיעוט מסויימות.

משפט: אלגורתם "קבלה על תנאי" שבו הסטודנטים מציעים הוא לא אמיתי עבור המחלקות.
משפט: אלגורתם "קבלה על תנאי" שבו הסטודנטים מציעים הוא כן אמיתי עבור הסטודנטים.

מנגנון קבלה על תנאי:

– כשהסטודנטים מציעים, אמיתי לסטודנטים אך לא למחלקות.
– כשהמחלקות מציעות, אמיתי למחלקות אך לא לסטודנטים.

משפט: לא קיים מנגנון לשידוך יציב שהוא אמיתי לשני הצדדים!

מטלה – שידוכים ושיבוצים

שאלה 1: שידוך יציב ויעילות

א. הוכיחו שכל שידוך יציב הוא יעיל-פארטו, אבל ההיפך לא נכון.
ב. נניח שכל סטודנט מייחס ערך מספרי לכל מחלקה, כאשר המחלקה הכי גרועה בעיניו מקבלת 1, המחלקה הבאה מקבלת 2, וכו' (אם יש m מחלקות אז המחלקה הכי טובה מקבלת m). גם כל מחלקה מייחסת ערך מספרי לכל סטודנט באותו אופן. האם כל שידוך יציב ממקסם את סכום הערכים? הוכיחו או הראו דוגמה נגדית.

תשובה 1:

כיוון ראשון: נניח השידוך יציב, נראה יעיל פארטו.
נניח בשלילה שיש שיפור פארטו. אזי בשיפור פארטו יש מישהו שבמצבו השתפר. בשיפור השחקן שמצבו השתפר משודך למחלקה שהיא יותר טובה עבורו, והמחלקה לא מפסידה (אחרת לא היה שיפור פארטו). לכן הסטודנט שמצבו השתפר והמחלקה המשודכת החדשה הם זוג מערער בשידוך הקודם, בסתירה לכך השידור יציב.
כיוון שני: נניח שהשידוך יעיל פארטו, נראה שהשידוך יציב.
נניח בשלילה שהשידוך לא יציב, אזי יש זוג מערערים, כך שאם נשדך ביניהם ערך השידוך יעלה, בסתירה לכך שהשידוך יעיל פארטו.

שאלה 2. תיכנון מערכת לשיבוץ

בעקבות תלונות על תהליך השיבוץ בין מתנדבות שירות-לאומי לבין מקומות-שירות, קיבלתם מכתב מרשות השירות הלאומי המבקש מכם לתכנן מערכת שיבוץ חדשה. המערכת אמורה לקבל קלט מהמתנדבות וממקומות-השירות, ולהחזיר כפלט את השיבוץ. הדרישות מהמערכת:
המערכת יכולה לשבץ כמה מתנדבות לכל מקום-שירות, בהתאם למספר התקנים במקום-השירות (מספר התקנים הוא בין 1 ל-100).
הפלט אמור להתקבל במהירות – דקות ספורות לאחר שכל המתנדבות ומקומות-השירות סיימו להקליד את הקלט.
לאחר שהתקבל שיבוץ – אף מתנדבת לא יכולה לשפר את מצבה על-ידי פניה טלפונית למקום-שירות שהיא מעדיפה על-פני זה ששובצה אליו, כי אף מקום-שירות כזה לא ירצה אותה. ולהיפך: אף מקום-שירות לא יכול לשפר את מצבו על-ידי פניה טלפונית למתנדבת שהוא מעדיף על-פני אלה ששובצו אליו, כי אף מתנדבת כזאת לא תרצה אותו.
בשלב הקלט, אף מתנדבת לא יכולה לשפר את מצבה על-ידי הכנסת קלט לא אמיתי.
תארו את המערכת בפירוט. יש להסביר את המערכת בעברית באופן שראש הרשות לשירות לאומי יוכל להבין. אין להשתמש בביטויים כמו "כפי שלמדנו בכיתה" כיוון שראש הרשות לא היה בכיתה...
א. מהו הקלט למערכת? מה הם מסכי-הקלט הדרושים, מי צריך להשתמש בכל מסך ומה בדיוק להקליד שם? תנו דוגמה לכל מסך.
ב. לאחר שכל המשתתפים סיימו להקליד את הקלט, איך בדיוק מתבצע האלגוריתם? הדגימו את פעולת האלגוריתם והפלט שלו על הקלט של סעיף א.
ג. הוכיחו שהאלגוריתם מקיים את תכונה 2 ע"י חישוב סיבוכיות זמן הריצה שלו.

תשובה 2:

המערכת תקבל מכל נערה דירוג למקומות ההתנדבות ומכל מקום התנדבות דירוג למתמודדות.
עבור כל נערה המערכת תבדוק האם יש מקום פנוי עבורה במקום ההתנדבות הבא ברשימת העדפותיה. אם יש מקום המערכת תכניס אותה לרשימה, ואם אין מקום המערכת תבדוק ברשימת

העדפות של מקום ההתנדבות האם יש משהי שהרשימה אבל בדורגת במקום פחות טוב מהנערה הנוכחית, במידה וכן יתבצע חילוף, אחרת לא יתבצע חילוף והנערה תנסה את מזלה במקום ההתנדבות הבא ברשימת העדפותיה. המערכת תמשיך עד שכל הנערות משובצות.

הקלט עבור המערכת יהיה:
עבור הנערות: שם,תעודת זהות, ורשימת העדפות עבור מקום ההתנדבות.
עבור מקומות ההתנדבות: שם, מספר המקומות שיש, ורשימת העדפות עבור הנערות.

שאלה 3: כמה שידוכים יציבים יש?
כשיש 3 סטודנטים ו-3 מחלקות, יש 6 שידוכים אפשריים (3 עצרת). כמה מהם יציבים?
א. הראו דוגמה שבה יש רק שידוך יציב אחד.
ב. הראו דוגמה שבה יש בדיוק 2 שידוכים יציבים.
ג. האם ייתכן שכל 6 השידוכים הם יציבים? אם כן - הראו דוגמה, אם לא – הוכיחו.

תשובה 3:
א.

	מקום 1	מקום 2	מקום 3
עמי	כלכלה	ספרות	היסטוריה
תמי	ספרות	היסטוריה	כלכלה
רמי	היסטוריה	כלכלה	ספרות

	מקום 1	מקום 2	מקום 3
כלכלה	עמי	רמי	תמי
ספרות	תמי	עמי	רמי
הסטוריה	רמי	רמי	עמי

שידוך יציב:
כלכלה – עמי
ספרות – תמי
הסטוריה- רמי

שאלה 4: מי נשאר בחוץ?
נניח שמספר הסטודנטים גדול ממספר המחלקות (ובכל מחלקה יש מקום אחד). במצב זה, בכל שידוך, חלק מהסטודנטים יישארו בחוץ.
א. הראו דוגמה עם 3 סטודנטים ו-2 מחלקות, שבה יש לפחות שני שידוכים יציבים. מיצאו את שניהם והראו איזה סטודנט נשאר בחוץ בכל שידוך יציב.
ב. הוכיחו, שב*כל* שידוך יציב (לא משנה באיזה אלגוריתם מצאנו אותו), *אותם* סטודנטים יישארו בחוץ. כלומר, אם מישהו נשאר בחוץ בשידוך יציב א, אז הוא נשאר בחוץ גם בשידוך יציב ב.
אם אתם נתקעים, קיראו בויקיפדיה על "Rural hospitals theorem".

תשובה 4:
סטודנטים:
עמי – כלכלה, מתמטיקה.
תמי–מתמטיקה,כלכלה.
בני– כלכלה, מתמטיקה.

מחלקות:
כלכלה– עמי, תמי, בני.
מתמטיקה– תמי, עמי, בני.

האלגוריתם:
עמי יפנה למחלקה למתמטיקה שתקבל אותו על תנאי, תמי תפנה למחלקה למתמטיקה שתקבל אותה על תנאי. בני יפנה תחילה למחלקה לכלכלה שתדחה אותו ולאחר מכן יפנה למחלקה למתמטיקה שתדחה אותו אף היא. האלגוריתם יסתיים ובני נותר ללא שידוך, שני השידוכים האחרים יציבים: עמי – כלכלה,תמי–מתמטיקה.

שאלה 5: **אלגוריתם קבלה–על–תנאי**
א. הראו דוגמה עם 3 סטודנטים ו–3 מחלקות שבו אלגוריתם קבלה על–תנאי מחזיר תמיד את אותו שידוך, בין אם הסטודנטים מציעים או המחלקות מציעות. תארו את ריצת האלגוריתם בכל אחד מהמצבים.
ב. הוכיחו שאלגוריתם קבלה על–תנאי כשהמחלקות מציעות, מחזיר את השידוך היציב *הכי גרוע עבור הסטודנטים* מכל השידוכים היציבים (רמז: הוכיחו שכל שידוך יציב שהוא הכי טוב עבור המחלקות, הוא הכי גרוע עבור הסטודנטים).
ג. באלגוריתם "קבלה על תנאי", כל מחלקה מדרגת את הסטודנטים בלי לראות מה הסטודנטים דירגו. נניח עכשיו שאנחנו מאפשרים למחלקות לראות את הדירוג של הסטודנטים לפני שהן מפרסמות את הדירוג שלהן. האם האלגוריתם עדיין אמיתי?

תשובה 5:
א.

	מקום 1	מקום 2	מקום 3
עמי	כלכלה	ספרות	היסטוריה
תמי	ספרות	היסטוריה	כלכלה
רמי	היסטוריה	כלכלה	ספרות

	מקום 1	מקום 2	מקום 3
כלכלה	עמי	רמי	תמי
ספרות	תמי	עמי	רמי
הסטוריה	רמי	רמי	עמי

שידוך יציב:
כלכלה – עמי
ספרות – תמי
הסטוריה– רמי

האלגוריתם כבר לא אמיתי, מכיוון שהדירוג של כל מחלקה כעת יתחשב בההעדפות של הסטודנטים. נגיד מחלקה רואה שסטודנט כלשהו שם אותה במקום האחרון בדירוג שלו, אז היא תעדיף לא לשים אותו בדירוג גבוהה גם אם היא הכי רוצה אותו, כי הסבירות שתקבל אותו נמוכה.

שאלה 6: **שידוך יציב לפי גובה**
בליגת כדורסל מסויימת, לכל שחקן יש העדפות שונות על הקבוצות, אבל לכל הקבוצות יש אותן העדפות על השחקנים – כל הקבוצות מדרגות את השחקנים מהגבוה לנמוך. כיתבו אלגוריתם לשידוך יציב בין קבוצות לשחקנים, שהוא פשוט יותר מאלגוריתם קבלה על-תנאי, אך עדיין אמיתי. הוכיחו שהוא אמיתי.

תשובה 6:
נמיין את כל השחקנים בסדר יורד של הגובה (מהגבוה לנמוך), כל שחקן בתורו יבחר את הקבוצה שהוא הכי מעדיף, במידה והקבוצה כבר מלאה, ינסה את הקבוצה הבאה בדירוג וכך האלה עד שכל השחקנים ישודכו.
האלגוריתם יעיל- כל שחקן רוצה בכל שלב נתון למצוא את הקבוצה הכי טובה שמתאפשר לו(שיש אצלה עדיין מקומות פנויים), אם השחקן יגיד את האמת זה בדיוק מה שהאלגוריתם עושה, משבץ אותו לקבוצה הכי טובה בעיניו שפנוייה.

שאלה 7: **תיכנות – יציבות**
כיתבו אלגוריתם המקבל שידוך ובודק אם הוא יציב.
בניגוד למטלות הקודמות, הפעם אתם תגדירו את מבני-הנתונים הדרושים, פורמט הקלט והפלט.

<u>אלגוריתמי החלפה (שיעור 10):</u>

דוגמאות:
- החלפת תורנויות בין עובדים.
- החלפת חפצים משומשים בקהילה.
- החלפת חדרים בין סטודנטים במעונות.

שאלה: למה לא להריץ פשוט את האלגוריתם לחלוקה הוגנת ?
תשובה: כי טודנטים שכבר יש להם חדרים יחששו להפסיד ויעדיפו לא להשתתף.

רציונליות ליחידים: אלגוריתם נקרא רציונלי ליחידים, אם מצבו של כל משתתף לאחר ביצוע האלגוריתם, טוב לפחות כמו מצבו לפני הביצוע.
כלומר, אף אחד לא מפסיד מההשתתפות, כולם משתתפים מרצונם האישי.

אלגוריתם מעגלי המסחר Top Trading Cycles:

האלגוריתם:
0. מאתחלים גרף מכוון שבו, הצמתים הם האנשים והבתים. יש קשת מכל אדם לבית שהוא הכי רוצה, ומכל בית לאדם שגר בו עכשיו.
א. מוצאים מעגל מכוון בגרף (DFS).
ב. מבצעים את ההחלפה במעגל.
ג. מוחקים מהגרף את הצמתים שהשתתפו בהחלפה.
ד. מעדכנים את הקשתות של האנשים שנשארו.
ה. חוזרים על שלבים א-ד עד שהגרף ריק.

משפט: אלגוריתם מעגלי המסחר מסתיים.
משפט: אלגוריתם מעגלי המסחר רציונלי.
משפט: אלגוריתם מעגלי המסחר הוא אמיתי.
משפט: אם כל יחסי ההעדפה הם חזקים (אין עדישות), אז אלגוריתם מעגלי המסחר יעיל פארטו.

קואליציה מערערת: קבוצת משתתפים שיכולה לפרוש ולבצע החלפת בתים שהיא טובה באותה מידה לכל חברי הקבוצה וטובה יותר לחלק מחבריה.
שיבוץ יציב: שיבוץ שאין בו קואליציה מערערת.

משפט: אם כל יחסי ההעדפה הם חזקים, אז מנגנון מעגלי המסחר מוצא שיבוץ יציב.
משפט: אם כל יחסי ההעדפה הם חזקים, אז יש רק שיבוץ אחד!

<u>החלפת כליות:</u>

כמעט בכל המדינות יש מחסור בכליות להשתלה, אסור לתרום כליות תמורת כסף, מותר לתרות כליה תמורת כליה.
שאלה: למה בכלל להחליף כליות ?
תשובה: תורם מוכן לתרום לחולה מסויים אך אין ביניהם התאמה מספקת בעקבות סוג דם שונה או סיבות אחרות נוספות. אם ישנה התאמה בים שני זוגות כאלה שביניהם אין להם התאמה אז ניתן לבצע החלפה ביניהם.

אלגוריתם שידוכי מסחר:
אלגוריתם מעגלי המסחר לא התאים לבעיה:
– המעגלים היו ארוכים מידי! בהחלפת כליות מעדיפים מעגלים קצרים באורך של 2 או 3. משום שההשתלות צריכות להתבצע במקביל.
– מצד שני, בהחלפות כליות ההעדפות הן בינאריות– כל חולה מוכל לקבל כליה מכל תורם מתאים.
הפתרון: במקום לחפש מעגלים, נחפש שידוכים.

שידוך בגרף כללי: אוסף של זוגות–צמתים זרים. כל צומת מייצג זוג, כל קשת מייצגת התאמה הדדית.

מסלול שיפור: מתחיל ומסתיים בצמתים לא משודכים, ובעל צבעים מתחלפים.

הלמה של ברג': שידוך M מקסימלי אם ורק אם אין מסלול שיפור.
איך מוצאים מסלול שיפור? אלגוריתם הפרחים, נלמד בקורס מתקדם יותר בתורת הגרפים.

האם האלגוריתם אמיתי?
מי הם השחקנים בבעיית שידוך הכליות?
– הזוגות, יכולים לכל היותר להסתיר קשתות, אבל זה לא יעזור להם.
– המרכזים הרפואיים, יכולים להסתיר זוגות, לשדך אותם באופן פנימי. האינטרס של המרכזים הרפאיים הוא לדאוג לחולים "שלהם", שכמה שיותר חולים שלהם יקבלו כליה.

משפט: אין אלגוריתם שהוא גם יעיל פארטו וגם אמיתי עבור המרכזים הרפואיים.

תמריצים של מרכזים רפואיים –קירוב ½:
כיוון שאין מנגנון אמיתי המשיג את השידוך הגדול ביותר, חיפשו מנגנון המשיג שידוך שהוא הגדול ביותר בקירוב.
המנגנון:
– מחשבים עבור כל מרכז רפואי, את המספר הגדול ביותר של קשתות פנימיות.
– מחשבים את השידוך הגדול ביותר מבין כל השידוכים עם אותו מספר של קשתות פנימיות.

<u>מטלה – החלפת בתים וכליות</u>

שאלה 1: החלפת בתים - יעילות
א. הוכיחו את המשפט: אם כל יחסי ההעדפה הם חזקים (אין אדישות), אז אלגוריתם החלפת הבתים תמיד מחזיר שיבוץ יעיל פארטו.
ב. הראו דוגמה שבה המשפט של סעיף א לא נכון, כאשר יחסי ההעדפה הם לא חזקים (כלומר יש אדם שהוא אדיש בין שני בתים).

תשובה 1:
א. נניח כל יחסי ההעדפה חזקים, נראה כי אלגוריתם החלפת הבתים תמיד מחזיר שיבוץ יעיל פארטו. נניח בשלילה שקיים שיפור פארטו. לפי ההנחה יש לפחות שחקן אחד שקיים עבורו שיבוץ יותר טוב. עבור שחקן זה האלגוריתם משבץ עבורו את הבית שהוא הכי רוצה –סתירה. לכן האלגוריתם מחזיר שיבוץ יעיל פארטו.
ב. דוגמה:
עמי גרה בבית א' ויש לה עדישות בין בתים א׳ ו–ב׳. תמי גרה בבית ב' אבל מעדיפה את בית א'. אם עמי יצביע על בית א'(כי האלגוריתם דורש ממנו לבחור העדפה אחת) אז האלגוריתם ישאיר אותו בבית שלו, ויוציא אותם מהמשחק, ובשלב השני ישאיר את תמי בביתה והאלגוריתם יסתיים. אבל קיים שיפור פארטו, אם האלגוריתם היה מבצע החלפה בין עמי ותמי, תמי הייתה מרוויחה ועמי לא היה מפסיד.

שאלה 2: החלפת בתים - תיכנות
כיתבו פונקציה, בשפת-תיכנות לבחירתכם או בפסאודו-קוד, המוצאת מעגל בגרף החלפת בתים.
כותרת הפונקציה:

```
vector<int> find_trading_cycle(vector<vector<int>> preferences);
```

הניחו שבתחילת הפונקציה, כל אדם i גר בבית i. הפרמטר preferences מתאר את ההעדפות של בעלי-הבתים. לכל i, הוקטור preferences[i] מתאר את סדר ההעדפות של אדם i. למשל, אם:
preferences[11] = {15, 13, 11, 7, 8, … }
המשמעות היא, שאדם מספר 11 הכי רוצה את בית 15, אחריו את בית 13, אחריו את בית 11 (הבית הנוכחי שלו), וכו'..
הפונקציה מחזירה וקטור המתאר את המעגל, למשל אם מוחזר הוקטור:
{11, 15, 17, 11}
המשמעות היא, שאדם 11 מקבל את בית 15, אדם 15 מקבל את בית 17, ואדם 17 מקבל את בית 11.

שאלה 3: החלפת כליות – שני אלגוריתמים

בשאלה זו נניח שהתאמה בין תורם לנתרם תלויה רק בסוג הדם. נתון מאגר-נתונים ובו שלושה זוגות עם סוגי-דם לפי הטבלה בצד שמאל.
א. ציירו את הגרף המכוון המתאר את ההתאמות בגרף.

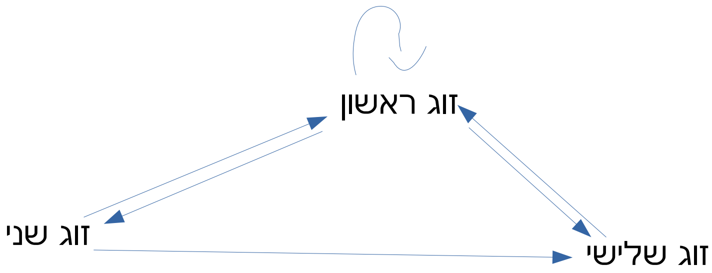
	סוג-דם חולה	סוג-דם תורם
זוג ראשון	AB	O
זוג שני	O	A
זוג שלישי	A	AB

ב. כמה ואיזה חולים אפשר להציל בלי החלפת כליות בכלל?
ג. כמה ואיזה חולים אפשר להציל בעזרת האלגוריתם למציאת שידוך גדול ביותר (שנלמד בכיתה)?

ד. כמה ואיזה חולים אפשר להציל בעזרת אלגוריתם למציאת שלשות רבות ביותר (שלא נלמד בכיתה) ?

תשובה 3:

א.



ב. ללא החלפת כליות ניתן להציל את הזוג הראשון בלבד.

ג. באמצעות האלגורתם שנלמד בכיתה ניתן להציל שני אנשים זוג ראשון ושלישי או זוג ראשון ושני.

ד. בעזרת אלגוריתם למציאת שלשות רבות ביותר ניתן להציל את כל שלושת האנשים.

שאלה 4: החלפת כליות בישראל

א. באיזה מרכזים רפואיים בישראל אפשר להירשם להחלפת כליות ?

ב. מהו האורך המקסימלי של מעגל-החלפה בתוכנית הישראלית ?

ג. באיזו תדירות מריצים את האלגוריתם לחיפוש מעגלים ?

ד. האם האלגוריתם הממומש בישראל הוא רציונלי-ליחידים (individually-rational) ? הוכיחו את טענתכם בעזרת שלוש טענות לפחות מתוך אתר משרד הבריאות.

תשובה 4:

א. ניתן להרשם להחלפת כליות ב:

1. מרכז רפואי רמב"ם (מבוגרים וילדים)- חיפה
2. מרכז רפואי איכילוב (מבוגרים) - תל אביב
3. מרכז רפואי בלינסון (מבוגרים) ושניידר (ילדים) – פתח תקווה
4. מרכז רפואי הדסה (מבוגרים) – ירושלים
5. מרכז רפואי סורוקה (מבוגרים) - באר-שבע

ב. במציאות האורך הכי גדול של מעגל החלפות בישראל היה בגודל 4.

ג. אחת לשלושה חודשים או שכאשר מצטברים לפחות 30 זוגות.

שאלה 5: החלפת כליות - תיכנות

כיתבו פונקציה, בשפת-תיכנות לבחירתכם או בפסאודו-קוד, המקבלת מטריצת התאמה בין זוגות, ומוצאת מעגלים **באורך 2 או באורך 3**, כך שמספר המושתלים הכולל הוא הגבוה ביותר. הפונקציה לא חייבת להיות יעילה חישובית (כלומר, אפשר לכתוב פתרון brute force). כותרת הפונקציה:

```
void find_max_matches(matrix<bool> compatible);
```

המטריצה compatible מתארת התאמה בין תורם לחולה. למשל compatible[11,22]==true אומר שתורם מספר 11 יכול לתרום לחולה מספר 22; compatible[22,11]==true אומר שתורם מספר 22 יכול לתרום לחולה מספר 11 (שימו לב – המטריצה לא בהכרח סימטרית).

פלט לדוגמה:

Length 2 cycle: 11→22 and 22→11

Length 3 cycle: 44→55 and 55→99 and 99→44

שיעור 11) Bitcoin and Blockchains:

מה זה בכלל כסף ולמה צריך אותו?
- מסחר בין שני אנשים יכול לשפר את התועלת של שניהם.
- סחר חליפין היא יעיל אבל דורש התאמה הדדית ברצונות. ראינו לכך דוגמאות, למשל אלגוריתם מעגלי המסחר להחלפת בתים.
- כסף משמש לתיווך במסחר.
- אדם נותן חפץ בתמורה לכסף, כי הוא מאמין שבעתיד, יוכל לתת את הכסף בתמורה לחפץ אחר.

תנאים הכרכיים למטבע:
1. קשה לייצר את המטבע (אחרת אף אחד לא יתן תמורתו חפצים, וכולם ייצרו בעצמם).
2. קל להוכיח שיש לי מטבע.
3. לא ניתן לשכפל, כלומר לשלם וגם להשאיר את המטבע אצלי.

התנאים הללו לא מספיקים, גם אם כל התנאים הללו מתקיימים, דרוש אמון במטבע.

זהב (או מתכות יקרות אחרות) כמטבע:
1. קשה לייצר, אפשר לכרות מהאדמה אבל זה לוקח המון זמן ומשאבים.
2. קל להוכיח, כולם רואים שאני מחזיק זהב.
3. אי אפשר לשכפל, כששילמתי הזהב כבר אינו אצלי.

שטרות מנייר:
1. קשה לייצר, יש אמצעים נגד זיוף.
2. קל להוכיח, כולם רואים שאני מחזיק שטר.
3. אי אפשר לשכפל, כששילמתי השטר כבר אינו אצלי.

מטבעות קריפטוגרפיים:

מטבע קריפטוגרפי מוגדר כאוסף של עסקאות.
1. קשה לייצר, דורש אמון.
2. קל להוכיח, דורש חתימות דיגיטליות.
3. אי אפשר לשכפל, דורש אלגוריתם.

מערכת חתימה דיגיטלית כוללת כמה רכיבים:
- אלגוריתם לייצור זוגות (מפתח סודי ← מפתח ציבורי)
- אלגוריתם חתימה בעזרת מפתח סודי.
- אלגוריתם אימות בעזרת מפתח ציבורי.

מפתח ציבורי, הוא כמו "שם משתמש" גלובלי.

איפה המטבע נמצא?

הגדרה: "מטבע קריפטוגרפי" הוא רשימה מקושרת-אחורה של הודעות מהצורה: "10 מטבעות נוצרו ונמסרו למפתח ציבורי א'", "א' שילם את 10 המטבעות שקיבל בהודעה הראשונה, למפתח ציבורי ב'".

מי שקורא את השרשרת, יכול לוודא תקינות על ידי אימות על ההודעות בעזרת המפתחות הציבוריים. כך אפשר לדעת למי שייך המטבע.

מניעת שיכפול:

הבעיה העיקרית ברשימה המקושרת היא- מוניעת תשלום כפול.

מניעת שכפול באמצעות הוכחת עבודה:

- הקונה שולח בקשת תשלום לרשת.
- כל משתמש בודק שהבקשה חוקית ומנסה לאשר.
- כדי לאשר בקשה m, צריך לפתור חידה מאוד קשה. להפוך פונקציה חד כיוונית, למצוא X כך ש:

$$SHA256(m + x) < (2^{224})/D$$

כאשר D הוא מספר המיצג את רמת הקושי.

הראשון שמצליח לפתור את החידה, שולח את הבלוק עם הפתרון לכולם וכך מצרפו לשרשרת.

שרשרת בלוקים-מושגים:

האישור מתבצע לא על כל עיסקה בנפרד, אלא על בלוקים של עסקאות (לכן Blockchain).

- בלוק מכיל Mb1 של עסקאות.
- כל עסקה תופסת בערך 0.5Kb.
- לכן בכל בלוק יש מקום לכ-2000 עסקאות.
- פתרון החידה (X) נקרא nonce.
- תהליך מציאת ה- nonce נקרא כריה.
- רמת הקושי (D) נקבעת באופן דינאמי כך שהזמן הדרוש למצואת ה nonce יהיה כ-10 דקות.

מה מרוויחים הכורים:

הכריה דורשת זמן והמון חשמל, מה יוצא להם מזה?

- דמי כריה קבועים.
- עמלת עסקה משתנה, נקבי על ידי המשלם בכל עקסה. כלל שהעמלה גבוהה יותר יש סיכוי גדול יותר שאחד הכורים יסכים להכניס את העסקה לבלוק.

מבנה של בלוק בשרשרת:

מזהה (HASH) של הבלוק הקודם.

כ- 2000 עסקאות. כאשר כל עסקה כוללת:

- מפתח ציבורי של שולח אחד או יותר.
- מפתח ציבורי של מקבל אחד או יותר.
- כמה כסף עבר מהשלוח למקבל.
- קישור לעסקאות קודמות שבהם התקבל הכסף.
- עמלת אישור

פתרון החידה המתאימה לתוכן הבלוק.

מפתח ציבורי של המאשר הזוכה בדמי הכריה.

מועדוני כריה:

לכורה בודד יש סיכוי מאוד קטן למצוא בלוק. שיתוף פעולה בין כמה כורים מגדיל את הסיכוי.

במועדון כריה העבודה של כל בלוק מתחלקת בין הרבה כורים. כשמישהו מוצא בלוק, המועדון לוקח את הכסף ומחלק אותו בין הכורים.

מחלוקות והתקפות:

<u>מזלגות ובלוקים יתומים</u>: ברגע שבלוק מאושר נשלח לרשת, כל שאר הכורים צרכים לזרוק את הבלוק שניסו לאשר, ולהתחיל לעבוד על בלוק חדש שבו ה"קישור לבלוק הקודם" הוא מזהה את

הבלוק החדש שאושר. אם שני כורים מאשרים בלוקים שונים במקביל, נוצר מזלג בשרשרת הבלוקים.
איך מחליט כל כורה, לאיזה בלוק קודם לקשר?
– כלל א: בחר את השרשרת הארוכה ביותר.
– כלל ב: אם יש כמה שרשראות ארוכות ביותר, קשר לבלוק ששמעת עליו קודם.
– בלוק שנמצא מחוץ לשרשרת הארוכה ביותר נקרא "יתום", העסקאות בו לא מאושרות.

<u>פיצולים מכוונים:</u> הקוד של ביטקוין הוא קוד פתוח, ויש ויכוחים בין המפתחים. כשלא מצליחים להכריע בויכוח, המטבע מתפצל. למשל, ב 1/8/17 היה פיצול בשרשרת ביטקוין בגלל ויכוח על גודלו של הבלוק. 1 מ"ב או להגדיל ל 8 מ"ג.

האם פרוטוקול הביטקוין אמיתי?
פרוטוקול נקרא אמיתי אם התנהגות בהתאם לפרוטוקול ממקסמת את הרווחים. התשלומים לכורים נועדו לעודד אותם לפעול לפי הפרוטוקול. אבל יש מקרים בהם משתלם לכורים לא לפעול לפיו:

<u>מתקפת תשלום כפול:</u> כורה המחזיק מעל ל 50% מכח הכרייה יכול לשלם פעמיים באותו בטבע באופן הבא: נניח שהבלוק הנוכחי הוא בלוק א'. התוקף קונה חפץ, והתשלום מאושר בבלוק ב' המקושר ל-א'. התוקף כורה בלוקים המקושרים לבלוק א', עד שהשרשרת שלו ארוכה יותר מהשרשרת העוברת דרך בלוק ב'. וכך בלוק ב' נעשה יתום, והאישור מתבטל!

<u>מתקפת קרטל הכורים</u>: מועדון כורים המחזיק שליש מכח הכרייה יכול להרוויח מהסתרת בלוקים שמצא. נניח שהבלוק הנוכחי הוא א', והקרטל מצא בלוק ב' המקושר אליו. במקום לפרסם את הבלוק, התוקף ממשיך לחפש בלוק ג' שיקושר לבלוק ב'. המתקפה גורמת לשאר הכורים לבזבז אנרגיה על בלוקים שבסוף יהפכו להיות יתומים. אפשר להוכיח שאם קרטל מחזיק לפחות 1/3, הרווח לכורה בקרטל גדולה מהרווח לכורה מחוץ לקרטל. לכן לכורים בודדים יש תמריץ להצטרף לקרטל שיגדל ויגדל עד שהוא יחזיק מעל ל 50% מכוח הכרייה.

<u>צליפת עמלות:</u> בעתיד הקרוב דמי הכרייה הקבועים יקטנו, ועמלת העסקה תהיה חלק משמעותי יותר מהשכר של הכורים. דבר שעלול להוביל להתנהגות חדשה ומתחכמת: נניח שבבלוק האחרון שנוסף לשרשרת, יש עיסקה עם עמלה מאוד גבוהה. לכורה יש תמריץ להמשיך את הבלוק הלפני האחרון, כדי שיוכל לשים בבלוק שלו את העסקה הזאת ולקחת את העמלה לעצמו. זה נקרא fee-sniping, אם כולם יעשו כך השרשרת תתיקע.

<u>שבירת שיוויון אנוכית:</u> לפי הפרוטוקול, במקרה של פיצול בשרשרת, כל כורה צריך להמשיך את הבלוק ששמע עליו ראשון. כשיש דמי כרייה קבועים זה לא משנה. אבל כשיש עמלות עסקה משתנות כדאי להמשיך דווקא את הבלוק שיש בו פחות עמלות, כי הוא משאיר יותר עמלות לכורים הבאים ומגדיל את הסיכוי ש "צלפי עמלות" ימשיכו אותך.

<u>חיתוך:</u> נניח שכמה כורים מבצעים שבירת שיוויון ונוכית. אז כדאי לכורים אחרים "לחתוך" את הבלוק הנוכחי ולהמשיך את הבלוק הקודם בבלוק שמכיל פחות עמלות, בתקווה שה"אנוכיים" ימשיכו אותך. כשכולם "חותכים" אחד את השני, נוצרת פרימת שוק,וכולם יוצרים בלוקים עם מעט מאוד עמלות.

כל המתקפות האלו עדיין לא נצפו בשטח כי העמלות עדיין נמוכות, אבל מה יהיה בעתיד???

מטלה – ביטקוין

שאלה 1: ארנקים ועסקאות – תרגיל מעשי

[ע"פ יואב הניג]

א. פתחו שני ארנקי testnet עפ"י ההוראות בסרטון: https://www.youtube.com/watch?v=LfNE29AZ9I0

ב. מיצאו "ברז" (faucet) עבור רשת הניסוי (testnet), והשיגו משם ביטקוין.

ג. העבירו קצת כסף מארנק 1 לארנק 2.

ד. מיצאו את העיסקה שלכם בשרשרת הבלוקים של רשת הניסוי, למשל כאן: https://live.blockcypher.com/btc-testnet

ה. כמה קלטים (inputs) יש בעיסקה שלכם? כמה פלטים (outputs)? מדוע?

ו. צרפו לפתרון השאלה, צילום מסך המראה את העיסקה שלכם בארנק ובשרשרת הבלוקים.

שאלה 2: קביעת רמת הקושי

לצורך השאלה, הניחו שקיימת המחלקה הבאה, המייצגת בלוק בשרשרת-הבלוקים:

```
struct Block {
    //... more methods
    long timestamp();     // time of block creation; seconds since 1/1/2008
    double difficulty();    // level of difficulty in block creation time.
    Block* previous();   // the preceding block in the chain.
    //... more methods
};
```

רמת הקושי של כריית בלוק צריכה להתעדכן בערך פעם בשבועיים. כיתבו פונקציה שאפשר להריץ פעם בשבועיים על-מנת להעריך את רמת הקושי הדרושה על-מנת שבלוק ייווצר בממוצע כל 10 דקות.

הפונקציה מקבלת קישור לבלוק האחרון בשרשרת, ומחשבת את רמת הקושי הדרושה בהתאם לקצב היצירה של הבלוקים מהשבועיים האחרונים. היעזרו בנוסחה הבאה, המחשבת את קצב היצירה הצפוי לפי רמת הקושי:

https://en.bitcoin.it/wiki/Difficulty#How_soon_might_I_expect_to_generate_a_block.3F

כותרת הפונקציה שאתם צריכים לממש:

```
double newDifficulty(Block* lastBlock);
```

תשובה 2: (הפתרון מועתק מהפתרון של אראל)

הזמן הדרוש לייצירת בלוק (T) שווה לרמת הקושי (D) כפול קבוע כלשהו (C):

T = D*C

במשך שבועיים צריכים להיווצר כ-2016 בלוקים (14 ימים כפול 144 בלוקים ליום). לכן, דרך פשוטה לחישוב רמת הקושי היא:

א. לחשב כמה זמן לקח ליצור את 2016 הבלוקים האחרונים (T0).

ב. לקרוא מהבלוקים את רמת הקושי האחרונה (D0).

ג. בהתאם לנוסחה שלמעלה, לחשב את הקבוע: C = T0/D0.

ד. הזמן ברצוי לייצירת 2016 בלוקים הוא שבועיים: T1 = 14* 86400.

ה. מכאן שרמת הקושי החדשה היא: D1 = T1/C = (T1/T0)*D0.

```
double newDifficulty(Block* lastBlock) {
    // a. calculate how much it took to create last 2016 blocks:
    long now = lastBlock->timestamp();
    for (int i=0, Block* b=lastBlock; i<2016; ++i)
        b = b->previous();
    long then = b->timestamp();
    long T0 = now-then;   // in seconds

    // b. read the most recent difficulty:
    long D0 = lastBlock->difficulty();

    // c. calculate the required time:
    long T1 = 14*86400;    // num of seconds in two weeks
    return (T1/T0)*D0;
}
```

שאלה 3: בדיקת תקינות עסקאות בשרשרת בלוקים

המטבע SimpleCoin הוא מטבע חדש (דמיוני), הדומה לביטקוין אבל הרבה יותר פשוט:
בכל בלוק בשרשרת יש רק עסקה 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 לכורֶה), ובלוק שבו מועבר מטבע שנוצר קודם. כל בלוק בשרשרת מיוצג ע"י המחלקה הבאה:

```
class Block {
        Block previous;
  // קישור לבלוק הקודם בשרשרת. אם הבלוק הנוכחי הוא הראשון //
 // בשרשרת (נוצר הכי מוקדם), אז השדה מכיל בלוק ריק – null. //
        Block input;
  // 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 //
     Key receiver;  // המפתח הציבורי שאליו מועבר המטבע בעיסקה זו.
};
```

עיסקה חדשה מיוצגת ע"י המחלקה הבאה:

```
class Transaction {
        Block input;  // ראו הסבר למעלה
     Key receiver; // ראו הסבר למעלה
     Signature signature;   // חתימה דיגיטלית של שולח העיסקה
};
```

חתימה דיגיטלית מיוצגת ע"י המחלקה הבאה:

```
class Signature {
    bool is_valid(Key signer);  // מקבלת כקלט את המפתח הציבורי של החותם
                                // מחזירה "אמת" אם"ם החתימה תקינה.
};
```

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

א. כיתבו פונקציה הבודקת האם עיסקה נתונה היא חוקית או לא:

```
void check(Transaction tx);
```

ניתן להניח שזו עיסקה של העברת מטבע ולא של יצירת מטבע חדש – כלומר tx.input!= null.
הפונקציה יכולה להסתיים באחת משלוש דרכים:

להדפיס "The transaction is valid";
לזרוק חריגה על "הוצאה כפולה" – DoubleSpendException ;
לזרוק חריגה על "חתימה לא חוקית" – IllegalSignatureException;

ב. ציירו שרשרת בת 4 בלוקים, והדגימו עליה את פעולת הפונקציה מהסעיף הקודם. תנו דוגמאות עם תוצאות שונות.

תשובה 3:
א.

```
void check(Transaction tx)
        Block b = latest
        while(b.previous in not null)
                if (b.input = =  tx.input) throw  DoubleSpendException
                b =  b.previous
        end-while
                Key =  ownerKey =  tx.input.receiver
                if(!tx.Signature.is_valid(ownerKey)) throw  IllegalSignatureException
        print("The transaction is valid")
end-check
```

שאלה 4: **חישוב יתרה**
כיתבו אלגוריתם המקבל כקלט מפתח ציבורי כלשהו, ומחשב את ה"יתרה" של בעל המפתח בביטקוין.

שאלה 5: **מתקפת פיני (על-שם** Hal Finney **שחשב עליה ראשון)**
נניח שמישהו, נקרא לו "התוקף", רוצה להשתמש במטבע אחד פעמיים. הוא פועל לפי האלגוריתם הבא. הוא כורה בלוקים באופן רגיל; בכל בלוק שהוא כורה, כאחת מ-2000 העיסקאות בבלוק, הוא מכניס עיסקה אחת של מטבע א מכתובת ב לכתובת ג, כאשר שתי הכתובות שייכות לו.
ברגע שהוא מצא בלוק, במקום לפרסם אותו מייד, הוא הולך לחנות, קונה חפץ ומשלם עליו במטבע א מכתובת ב לכתובת של המוכר. ברגע שהוא יוצא מהחנות, הוא שולח את הבלוק המאושר

שמצא בצעד הראשון. הבלוק מאושר, העיסקה שעשה עם המוכר נדחית, המוכר מפסיד את הכסף, והתוקף קיבל חפץ בלי לשלם.

א. מה הסיכון שהתוקף לוקח? באיזה מקרה ההתקפה תיכשל, וכמה יפסיד התוקף במקרה זה?

ב. המוכר מחכה t דקות לפני מסירת החפץ. מה צריך להיות ערך החפץ כך שההתקפה תהיה כדאית?

ג. מה יכול המוכר לעשות כדי להגן על עצמו מהתקפה זו?

אתריום וחוזים חכמים Ethereum and Smart Contracts (שיעור 12):

שרשראות בלוקים:

המצאת הביטקוין הביאה לעולם שני חידושים:
- מטבע מבוזר, לא תלוי בממשלה או בבנק.
- הסכמה מבוזרת על סדר אירועים.

בעבר התלהבו בעיר מהחידוש הראשון, כיום מהשני. שרשרת הבלוקים מאפשרת להסכים על אירועים ולבצע פעולות רבות שאינן קשורות דווקא למטבעות.
זהו הרעיון מאחורי מערכת אתריום.

חוזים חכמים:

כיום כדי לאכוף חוזה, דרוש גוף מרכזי. מערכת ארתיום היא מערכת שבה אפשר לכתוב חוזים חכמים, חוזים הניתנים לתיכנות. המערכת תאכוף אוטומטית על ידי ביצוע הקוד.
דוגמה: מכירת יצירת אומנות במכרז מחיר ראשון, כל משתתף שולח את הכסף לחוזה החכם. החוזה מחשב מקסימום ומגלה מי הזוכה. החוזה מחזיר את הכסף למפסידים.
דוגמות נוספת: ביטוח, למשל מפני ירידת שער. אם השער עולה/יורד, הלקוח מקבל את הכסף מיידית. שכירת חדר עם מפתח דיגיטלי. משחקים על כסף למשל פוקר.

מכונת אתריום וירטואלית EVM:

המושג הבסיסי-חשבון:
- חשבון בבעלות חיצונית (בדרך כלל בבעלות אדם).
- חשבון של חוזה. מייצג חוזה חכם בתוך המערכת. כולל קוד וזיכרון.

לכל חשבון, חיצוני או חוזה יש יתרה וכתובת.
היתרה היא מטבע הנקרא אתר (Ether).
אתר = 1000 פיני
= 1e9 גיגה-ווי.
הכתובת מאפשרת לקבל אתר והודעות.

מכונה אתריום וירטואלית-מצב:

המצב של ה EVM, בכל רגע נתון, כולל את:
- היתרה של החשבונות במערכת.
- הקוד והזיכרון של כל החוזים החכמים.

המצב של ה EVM נשמר בשרשרת הבלוקים. על כורה המעוניין להשתתף בשרשרת אתריום, צריך להוריד את כל שרשרת הבלוקים, לבצע את כל הקוד, ולחשב את המצב הנוכחי של ה-EVM.

מכונה אתריום וירטואלית-דלק:

כל פעולה שמריצה קוד, צורכת "דלק".
לכל פעולה בשפת EVM יש "צריכת דלק".
צריכת הדלק של עסקה היא 21000. צריכת דלק של כל בייט נוסף בעסקה הוא 68.
כששולחים פעולה לביצוע, צריך להגדיר כמות דלק מקסימלית ותשלום לכל יחידת דלק.
מחיר השוק לדלק כיום: 4 גיגה-ווי.
כשנגמר הדלק, הביצוע נפסק (לולאה אינסופית לא תתקע את הרשת).

איך מתחילים:

1. מתקינים תוכנת ארנק- תוכנה המנהלת מפתחות פרטיים וציבוריים, שולחת עסקאות.
2. פותחים חשבון אחד או יותר דרך הארנק. (חשבון הוא זוג מפתחות. נשמר מקומי בלבד).

3. משיגים קצת אתר, למשל, על ידי "ברז" של רשת ניסיונית כלשהי.

איך מתכנתים?

1. כותבים קוד בשפה ייעודית כלשהי כגון:

Solidity — שפה דמויית ג'אבה, Serpent — שפה דמויין פייתון.

2. מקמפלים לשפת המכונה של ה EVM:

– על מחשב מקומי solc, מסובך.

– בדפדפן, remix.ethereum.org, פשוט.

3. שולחים את החוזה המקומפל לרשת אתריום. דרך MetaMask, כמו כל עסקה אחרת.

מרגע שהחוזה נשלח, הוא לא ניתן לשינוי.

איך מפעילים?

1. שולחים לכל המשתמשים את קוד המקור של החוזה, הכתובת שבה החוזה נמצא ברשת.

2. כל משתמש יכול דרך remix:

– לקמפל את החוזה.

– להתחבר לחוזה הקיים.

– להפעיל את השיטות הציבוריות של החוזה.

– לראות את המשתנים הציבוריים של החוזה.

3. החוזה יכול לצבור כסף ולשלם כסף.

סיכונים:

באג בקוד של חוזה חכם, גלוי מייד לכל העולם, ויכול לעלות הרבה מאוד כסף!

אתגרים:

חוזים חכמים המסתמכים על אירועים חיצוניים צריכים לקבל מידע על העולם החיצון. למשל: חוזה ביטוח דירה, צריך לדעת האם הדירה נשרפה, האם הדייר נכנס לדירה.

פתרונות: חיישנים חכמים עם חשבון באתריום, מנעולים חכמים.

מטלה – אתריום

שאלה 1: אבן נייר ומספריים עם סודיות
א. ודאו שאתם יכולים לקמפל ולהריץ את החוזה של "אבן נייר ומספריים" שהוצג בכיתה (בקובץ code/rps.sol) בסביבת רמיקס: https://remix.ethereum.org (אמור לעבוד בדפדפן Chrome).
ב. שנו את התוכנית בהתאם למה שנלמד בהרצאה, כך שהבחירות של שני השחקנים יישמרו בסוד עד לסיום המשחק, ולא יהיה אפשר לראות אותן דרך שרשרת הבלוקים.

שאלה 2: משאל עם
כיתבו חוזה חכם בשפת solidity המאפשר לבצע משאל-עם. למשאל יש שתי תוצאות אפשריות: "כן" (1) או "לא" (0). החוזה יתמוך בפעולות הבאות:
addVoter – הוספת כתובת ציבורית של אזרח שיש לו זכות בחירה. שיטה זו ניתנת להרצה רק ע"י מי שיצר את החוזה.
vote – הצבעה – 0 או 1. שיטה זו ניתנת להרצה רק ע"י אזרח שנוסף לפני-כן לחוזה ע"י addVoter. כל אזרח יכול להצביע רק פעם אחת לכל היותר.
outcome – שיטה המחזירה את תוצאת המשאל – 0 או 1.

שאלה 3: מכרז
כיתבו חוזה חכם בשפת solidity המאפשר לבצע מכרז מחיר שני. בהתחלה המכרז הוא ב"בעלות" מי שיצר את החוזה; בכל פעם שמישהו זוכה במכרז, המכרז עובר לבעלותו. החוזה יתמוך בפעולות הבאות:
bid – שליחת כסף לחוזה לצורך השתתפות במכרז. ההכרזה שווה לכמות הכסף שנשלח.
conclude – סיום המכרז הנוכחי. שיטה זו ניתנת להרצה רק ע"י הבעלים הנוכחי של המכרז. השיטה מבצעת את הפעולות הבאות: (1) חישוב הזוכה – זה ששלח הכי הרבה כסף דרך bid. (2) החזרת הכסף לכל המשתתפים שלא זכו. (3) החזרת חלק מהכסף לזוכה – ההפרש בין המחיר שלו לבין המחיר השני. (4) העברת המחיר השני לבעלים הנוכחי של החוזה. (5) שינוי כתובת ה"בעלים" של החוזה לכתובתו של הזוכה – כך שהזוכה הופך להיות הבעלים החדש של החוזה.

שאלה 4. חוזה חכם לפתרון בעיה NP-שלמה
ב**בעיית 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 (להזכירכם, 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.

תשובה 4:
צריך לבדוק שהמערכים שהוחזרו באמת תואמים למעריך המקורי. כלומר, שפתר באמת את הבעיה שניתנה ולא סתם החזיר שני מערכים עם סכום שווה. בנוסף צריך לבדוק גם האם סכום של המערכים זהה. שכל בעיה נפתרת רק פעם אחת.

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

    function solve(int[] part1, int[] part2) public {
        int sum1 = findSum(part1)
        int sum2 = findSum(part2)
        if(sum1 != sum2) return wrong answer
        int[] merge = new arr [part1.length + part2.length]
        merge ← part1 , part2 //insert part1 and part2 to merge.
        sort(merge)
        sort(_input)
        if(merge.length != _input.length)return wrong answer
        for(j= 0 to _input.length)
            if(merge[j] != _input[j]) return wrong answer
        end-for
        send to solver 1 ether
        destroy contract
    }
}
```