מגישה: שגית מלכה

א. [הקוד בגיט]

דוגמאות הרצה:

1. קלט:

```
total_budget = 100
citizen_votes = [[100, 0, 0], [0, 0, 100]]
result = compute_budget(total_budget, citizen_votes)
print([round(x) for x in result])
```

פלט:

```
[50, 0, 50]
```

2. קלט:

```
result2 = compute_budget(60, [
[60, 0],
[60, 0],
[0, 60]])
```

פלט:

```
[40, 20]
```

3. קלט:

```
result3 = compute_budget(30, [
[10, 10, 10],
[0, 15, 15],
[15, 15, 0]])
```

פלט:

```
[10, 10, 10]
```

ב.

תיאור האלגוריתם:

נשתמש באלגוריתם הפנטומים הזזים עם פונקציות פנטום ליניאריות:

$f_i(t) = C * min(1, it)$ אבל במקום חיפוש בינארי על t, נשתמש בעובדה שהתקציה המתקבל הוא פונקציה קטעית-לינארית של t.

הרעיון המרכזי הוא שהחציון בכל סעיף יכול להשתנות רק כאשר פנטום $f_i(t)$ עובר ערך הצבעה של אזרח, לכן קיימים רק ערכי t סופיים שהבם סדר ההצבעות משתנה.

האלגוריתם:

1. מחשבים את כל ערכי המעבר: עבור $i = 1,\dots,n-1$

$t_{i,j,a} = \frac{v_{a,j}}{C \cdot i}$ ושומרים רק ערכים בתחום $[0, 1]$

2. ממיינים את כל הערכים ומקבלים $0 = t_0 < t_1 <... < t_k = 1$
3. עבור כל קטע $[t_i, t_{i+1}]$:
   a. מחשבים את סכום התקציב cקצוות הקטע
   b. אם הסכום שווה לC באחד מהקצוות - זה הפתרון
   c. אם הסכום חוצה את C בתוך הקטע - פותרים משוואה ליניארית ומוצאים $t *$
4. מחשבים את התקציב עבור $t *$

המימוש בפייתון כאן

הוכחת נכונות:

נציג כמה טענות:

1. מספר המצבים הוא סופי: החציון בכל סעיף משתנה רק כאשר פנטום חוצה הצבעה של אזרח. מספר ערכי־המעבר הוא $O(n^2 m)$ ולכן האלגוריתם מחשב מספר סופי של קטעים.
2. מבנה התקציב בתוך קטע: בין כל שני ערכי מעבר עוקבים סדר ההצבעות קבוע והחציון בכל סעיף הוא קבוע או פונקציה לינארית של t, לכן סכום התקציב הוא פונ׳ לינארית: $S(t) = \alpha t + \beta$
3. מציאת $t *$: אם $S(t_0) \leq C \leq S(t_1)$, אז קיימת נקודה יחידה $t * \in [t_0, t_!]$ כך ש $S(t *) = C$ והיא מחושבת במפורש ע״י פתרון משוואה ליניארית.

כלומר: האלגוריתם בוחם את כל הקטעים האפשריים, מוצא בוודאות $t *$ שבו סכום התקציב שווה ל C ומחזיר את התקציב של אלגוריתם הפנטומים הזזים, ולכן הוא מחזיר תשובה נכונה (ופולינומי).