1 ראשית האלגוריתם עוצר שכן קורה אחד מבין השניים בכל איטרציה של הלולאה: או שערכו של P משתנה ל-**False** או שמונה הלולאה i גדל באחד. במקרה הראשון הלולאה תעצור והאלגוריתם יסתיים, במקרה השני הלולאה תעצור לאחר $n/2$ איטרציות.

טענות ביניים:

1. לפני שורה 1: המערך B מכיל מחרוזת תווים באורך n.
2. לפני כל איטרציה של הלולאה בשורה 2: התו ה-$i-1$ שווה לתו ה-$n-i+2$ אם"ם $P=$ **True**.
3. אחרי היציאה מהלולאה: B פלינדרום אם"ם $P=$ **True**.

מעברים:

1-2. לפני תחילת האיטרציה הראשונה, ערכו של i הוא 1 ולא קיים תו במקום ה-0 או במקום ה-$n+1$ ולכן 2 מתקיים באופן ריק.

2-2. בתחילת האיטרציה ה-j מתקיים $P=$ **True** ו-$B[j-1]=B[n-j+2]$. אם התווים $B[j],B[n-j+1]$ שווים אז ערכו של P נשמר גם לאיטרציה הבאה.

2-3. אם ערכו של P נשמר, אז לפי 2 לכל $i=1,..,n/2$ התקיים $B[i]=B[n-i+1]$ ולכן B פלינדרום.

2 א רצים על המערך מהסוף ובכל איטרציה סוכמים את האיברים מתחילת המערך עד האיבר הנוכחי.

```
for i ← n to 2 do
    sum = 0
    for j ← 1 to i do
        sum += A[j]
    A[i] = sum
```

2 ב האיבר ה-i במערך החדש שווה ל-$\sum_{j=1}^{i} A[j]$, אם כך קל לראות שאת האיבר הבא במערך החדש נוכל לקבל כך: $A[i]+A[i+1]$.

```
for i ← 2 to n do
    A[i] += A[i − 1]
```

2 ג האינווריאנטה שמתקיימת לפני האיטרציה ה-i-ית היא ש-$A[i]=\sum_{j=1}^{i} A[j]$.

3 א .

```
procedure TernaryBinarySearch(A, p, q, v)
    if p > q then
        return False
    third ← (q − p)/3
    if A[q − third] = v or A[p + third] = v then
        return True
    else if v > A[q − third] then
```

```
8:        return TernaryBinarySearch(A, q − third + 1, q, v)
9:    else if v < A[p + third] then
10:       return TernaryBinarySearch(A, p, p + third − 1, v)
11:   else
12:       return TernaryBinarySearch(A, p + third + 1, q − third − 1, v)
```

3 ב נוכיח את הנכונות החלקית של האלגוריתם הנ"ל.

טענה: השגרה $\textsc{TernaryBinarySearch}$ מחזירה אמת אם"ם הערך v נמצא ב-A.

הוכחה: נסמן את אורכו של A ב-n.

בסיס האינדוקציה: אם $n = 1$ אז $third$ יקבל את הערך 0 ובשורה 6 יוחזר אמת אם האיבר היחיד ב-A שווה ל-v. אחרת בהתאם לערכו של $A[1]$ נבצע קריאה רקורסיבית באחת מהשורות 8, 10 ו-12 כאשר ערכו של p יגדל, או שיקטן q ולכן התנאי בשורה 2 יתקיים ויוחזר שקר.

צעד האינדוקציה: נניח שהטענה נכונה לכל $k < n$ ונראה שמכך נובעת נכונות הטענה עבור n.

האלגוריתם מחלק את המערך $A[p..q]$ לשלושה איזורים:

$$A[p..p + third], A[p + third + 1..q - third - 1], A[q - third + 1..q]$$

תחילה בודקים אם האיבר שאותו מחפשים נמצא "בתפר" שבין האיזור הראשון לשני או בין האיזור השני לשלישי ובמידה וכן נחזיר אמת.

אחרת תתבצע קריאה רקורסיבית לאחד משלושת תתי המערכים שהוזכרו לעיל, שכאמור גודל כל אחד מהם קטן פי 3 מגודלו של $A[p..q]$. מהנחת האינדוקציה נקבל שהערך שיוחזר מהקריאה הרקורסיבית ל-$\textsc{TernaryBinarySearch}$ יהיה אמת או שקר בהתאם להימצאו של v ב-A.

זמן הריצה של האלגוריתם מקיים את הנוסחת הנסיגה:

$$\begin{aligned} T(1) &= O(1), \\ T(n) &= T(n/3) + O(1) = T(n/9) + 2 \cdot O(1) \\ &= \cdots = T(n/3^i) + i \cdot O(1) \end{aligned}$$

הרקורסיה תיעצר לאחר $\log_3 n$ קריאות כי $T(n/3^{\log_3 n}) = T(n/n) = T(1) = O(1)$.
לסיכום נקבל ש- $T(n) = O(\log_3 n)$.

4 בכדור הראשון נבצע זריקה בינארית™ ב-n הקומות, כלומר נזרוק מהקומה ה-$n/2$, אם נשבר נבצע אלגוריתם שיתואר בהמשך למציאת הקומה המינימלית ב-$n/2$ הקומות הראשונות כשיש בידנו שני כדורים. במידה ולא נשבר הכדור, נמשיך ב-$n/2$ הקומות העליונות ע"י זריקת הכדור מהקומה ה-$n/4$, וכך הלאה.

בהנחה שזמן הריצה של האלגוריתם למציאת הקומה המינימלית עם 2 כדורים רץ ב-$T(n)$, אזי מבחינת סדר גודל אין שינוי, שכן במקרה הגרוע הכדור הראשון ישבר בזריקה הראשונה, אבל בכל זאת חצינו את המרחב בחצי לפחות.

נותר לתאר את האלגוריתם למציאת הקומה המינימלית עם 2 כדורים ולמצוא את $T(n)$. אם נזרוק את הכדור בקפיצות של 2, כלומר בקומות $2, 4, 6..$ וברגע שנשבר "נטפס" מהזריקה האחרונה אחד אחד, נגלה שבמקרה הגרוע הכדור ישבר בקומה ה-n (ואז ידוע שהקומה המינימלית היא $n-1$), כלומר סה"כ היינו צריכים $n/2$ זריקות. אם ננסה למצוא "קפיצה"

אופטימלית שכזו שתמזער את מספר הזריקות, נקבל את הפונקציה $f(x) = \frac{n}{x} + x - 1$ שמחזירה את מספר הזריקות כפונקציה של גודל הקפיצה בכל שלב. ע"י גזירה מקבלים מינימום ב-$\sqrt{n}$. כלומר אם נקפוץ ב-$\sqrt{n}$ בכל שלב, נצטרך במקרה הגרוע $O(\sqrt{n})$ זריקות.

אפשר להקטין עוד יותר את מספר הזריקות ע"י הקטנת גודל הקפיצה באחד בכל שלב ומציאת k מינימלי המקיים $1 + \cdots + k > n$, אבל גם פה מקבלים ש-$c = O(\sqrt{n})$.

לסיכום, ניתנחו אלגוריתם שבהינתן 2 כדורים ו-n קומות, מוצא את הקומה המינימלית בסיבוכיות של $O(\sqrt{n})$. נשתמש בו לפתרון הבעיה המקורית לאחר שכדור 1 נשבר. סה"כ הפתרון המוצע רץ ב-$O(\sqrt{n})$.

5 א 7 ו-8 הם האיברים היחידים שכל מה שמשמאלם קטן מהם וכל מה שמימינם גדול מהם.

5 ב מאתחלים מערך C שיכיל את כל הערכים המשותפים. רצים על A מההתחלה ועל B מהסוף. בכל איטרציה בודקים אם האיבר הנוכחי ב-A שווה לאיבר הנוכחי ב-B. במידה וכן והוא לא נמצא כבר ב-C, מוסיפים אותו. אם אין שיוויון והאיבר של B גדול יותר, עוברים לאיבר הבא ב-A, אחרת לאיבר הקודם ב-B. בכל איטרציה אחד מהאינדקסים זז קדימה או אחורה ולכן מתישהו הם יחצו אחד את השני והלולאה תסתיים.

```
C = n sized array initialized with nils
a = 1, b = n, c = 1
while a ≤ b do
    if A[a] = B[b] then
        if C[c] ≠ A[a] then
            if c > 1 then
                c += 1
            C[c] = A[a]
        a += 1
        b −= 1
    else if B[b] > A[a] then
        a += 1
    else
        b −= 1
```

5 ג רצים על המערך ומסתכלים על כל האיברים משמאל ומימין לאיבר הנוכחי. אם כל אלו שמשמאל קטנים שווים מהאיבר הנוכחי וכל אלו שמימין גדולים שווים אז מדפיסים את האיבר הנוכחי.