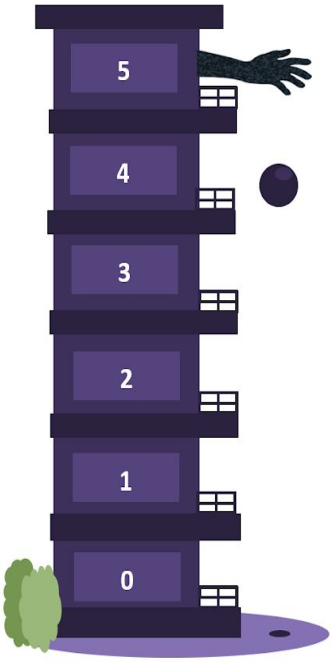
בעיית כדור הזכוכית



תיאור הבעיה

- ⇚ לפניכם בניין בן **n** קומות ולרשותכם **b** כדורי זכוכית.
- ⇚ עליכם לגלות, באמצעות זריקת הכדורים מהקומות, מהי הקומה הנמוכה ביותר שאם תזרקו ממנה כדור היא תישבר.
- ⇚ מובן שאם הכדור שזרקתם נשברה, לא תוכלו להשתמש בה שוב להמשך הבדיקה.

הערות -

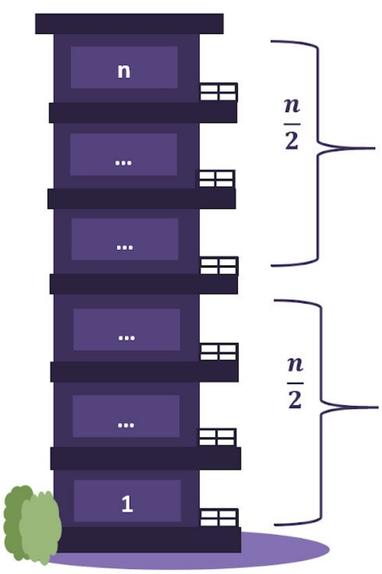
1. אם הכדור נשבר מקומה כלשהי, הוא גם יישבר מכל הקומות הגבוהות יותר.
2. אם הכדור לא נשבר לאחר זריקה, נשתמש בו שוב.
3. אם הכדור נשבר מקומה i אז היא תשבר גם בכל קומה i+1.
4. אם הכדור לא נשבר מקומה i אז היא לא תישבר גם בכל קומה i-1.
5. אם אני למשל עם 100 כדורים וקומה אחת בלבד, אז מספיק לי כדור אחד בשביל הבעיה.

פתרון הבעיה

❖ **בידינו b=1 כדורים בלבד, ו-n קומות.**

⇚ מתחילים את זריקות הכדור מקומה אחת – אם הוא יישבר, מצאנו את הקומה, במקרה שהכדור לא ישבר זורקים אותו מקומה 2 .וכך עולים קומה-קומה וזורקים את הכדור עד שהוא ישבר.

↜ **סיבוכיות זמן ריצה**: $O(n)$ - במקרה הגרוע.



❖ **בידינו b=2 כדורים בלבד, ו-n קומות.**

⇚ נחלק את הבניין ל**שני חלקים שווים**, זורקים כדור ראשון מקומה $\frac{n}{2}$:

אם הכדור הראשון נשבר, נתחיל מהקומה הראשונה ונעבור

בכל

קומה עד שישבר.

אם הכדור הראשון לא נשבר, נבדוק מקומה $\frac{3n}{4}$ וכן הלאה..

↜ **במקרה הטוב** - $log_2 n$ ניסיונות. שזה בעצם מקרה שיש לנו $log_2 n$ כדורים או יותר.

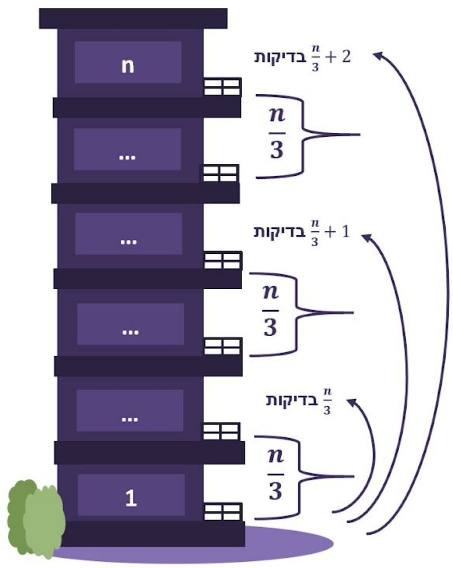
↜ **במקרה הגרוע** - יש לנו $1 + \frac{n}{2}$ ניסיונות (כי אם הכדור הראשון

נשבר, נשאר לי עוד $\frac{n}{2}$ ניסיונות).

⇚ נחלק את הבניין ל**שלושה חלקים שווים,** זורקים כדור ראשון מקומה $\frac{n}{3}$:

אם הכדור הראשון נשבר, נתחיל מהקומה הראשונה ונעבור בכל קומה עד שישבר.

אם הכדור הראשון לא נשבר, נבדוק מקומה $\frac{2n}{3}$ וכן הלאה…



סך הכל: $\frac{n}{3}+2$ ניסיונות (כפי שאפשר לראות בציור ניסינו בקומות הראשונות $\frac{n}{3}$ ולא נשבר, קפצנו 1+ , ניסינו $\frac{n}{3}+1$ לא נשבר, קפצנו עוד 1+ , ולכן במקרה הגרוע ניסינו $\frac{n}{3}+2$).

ראינו שאם נחלק ל**שני חלקים שווים**, במקרה הגרוע - $\frac{n}{2}+1$ ניסיונות.
ראינו שאם נחלק ל**שלושה חלקים שווים,** במקרה הגרוע - $\frac{n}{3}+2$ ניסיונות.
לכן, אם נחלק את הבניין ל-**k חלקים שווים**, במקרה הגרוע יש $\frac{n}{k}+(k-1)$ ניסיונות.
ואם נחלק את הבניין ל-**n חלקים שווים**, במקרה הגרוע יש $\frac{n}{n}+(n-1)$ ניסיונות.

נרצה לדעת מהי החלוקה האופטימלית ביותר

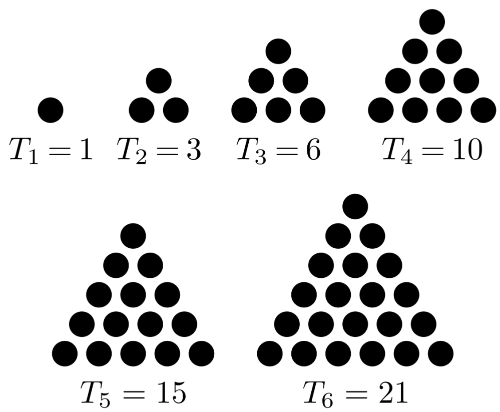
תזכורת - כדי לחשב מינימום של קטע, נגזור את הקטע, נשווה לאפס ואז נמצא את הערך המינימלי שלו.
לכן, מהו הערך של k שנותן מינימום לפונקציה $f(n,k)=\frac{n}{k}+(k-1)$, כאשר n הוא קבוע.
נחשב $min\, f(x) \;=\; (\frac{n}{x}+x)$, לכן נגזור:

$$f\,'(x)= \;-\frac{n}{x^2}+1=0$$
$$-\frac{n}{x^2}+1=0 \setminus +\frac{n}{x^2}$$
$$1=\frac{n}{x^2} \setminus \cdot x^2$$
$$x^2=n \setminus \cdot \sqrt{}$$
$$x=\sqrt{n}$$

לכן, קיבלנו שהחלוקה האופטימלית היא $\sqrt{n}$ חלקים שווים.
ובמקרה הגרוע כאשר $f(\sqrt{n})=\;\frac{n}{\sqrt{n}}+(\sqrt{n}-1)\;\approx 2\sqrt{n}$.

אם n הוא מספר ריבועי, אז $k=\sqrt{n}$.
אם n הוא לא מספר ריבועי, אז ניקח מספר ריבועי מינימלי m הגדול מ-n כך שמתקיים: $(m-1)^2\le n\le m^2$.

❖ **בידינו עדיין b=2 כדורים בלבד, ו-n קומות**, אבל נציג חלוקה אחרת - מספרים משולשים.
המספר המשולשי ה-n-י מסומן T_n , והוא שווה לסכום כל המספרים הטבעיים מ-1 עד n.
ישנם אינסוף מספרים משולשיים וניתן להציג כל מספר משולשי בצורת משולש שווה-צלעות.



נניח ש-n הוא מספר משולשי: $n = 1 + 2 + ... + k = \frac{k\cdot(k+1)}{2}$ **(נוסחת גאוס)** ,
נחלק את הבניין ל-k חלקים: (1,2,3,4...ועד k):
כאשר זורקים את הכדור הראשון מקומה k **והוא נשבר**, נשארים לנו עוד k-1 ניסיונות.
כלומר, סך הכל - k ניסיונות.
כאשר זורקים את הכדור הראשון מקומה k **והוא לא נשבר**, זורקים אותו מקומה $k + (k-1)$,
אם הוא נשבר, נשאר עם k-2 ניסיונות.
כלומר, סך הכל שוב - $k - 2 + 2 = k$ ניסיונות.

- תמיד יהיה לנו k ניסיונות.

כאשר n מספר משולשי ו- $n = \frac{k\cdot(k+1)}{2}$ אז:

$$2n = k(k+1)$$
$$2n = k^2 + k$$
$$2n - k = k^2$$

ומכיוון ש- $2n - k < 2n$ נקבל: $k^2 < 2n$, נוציא שורש ונקבל ש- $k < \sqrt{2n}$.

מסקנה - $\sqrt{2n} \leq 2\sqrt{n}$, ולכן חלוקה לפי מספר משולשים טובה יותר מחלוקה למספרים שווים - $\sqrt{n}$.
כי הראנו שמקרה הגרוע עבור חלוקה למספרים שווים נקבל $2\sqrt{n}$, ואם מקרה זה יכול להיות גדול יותר מהחלוקה המינימלית שמצאנו עבור מספרים משולשים - $\sqrt{2n}$ אז ברור שעדיף לנו להשתמש בשיטת המספרים המשולשיים.

❖ **בידינו b>2 כדורים , ו-n קומות**.

- בהינתן b כדורים ובניין בעל n קומות, נגדיר את הפונקציה $f(n,b)$ - מספר מינימלי של ניסיונות במקרה הגרוע:

$$f(n,2) \;=\; \min_{1\le i\le n}\; max(\,f(n-i,2)\;,\;f(i-1,1)\;)\;+1$$

- כאן משתמשים בתכנות דינמי מורכב יותר: בכל שלב משתמשים בכל התוצאות של השלבים הקודמים.

כאשר זורקים כדור מקומה i יש שתי אפשרויות:

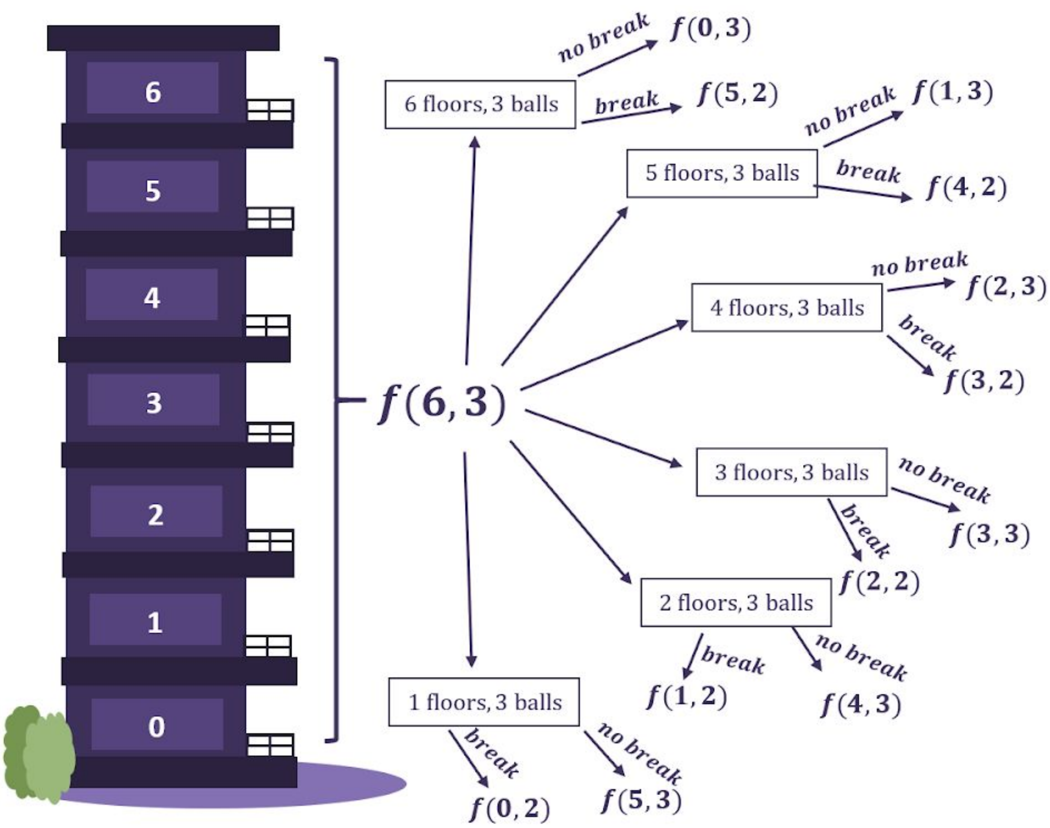
1. הכדור לא נשבר, נשארו 2 כדורים ו- $n-i$ קומות.
2. הכדור נשבר, נשאר כדור אחד ו- $i-1$ קומות. הולכים על המקרה הגרוע ומחשבים את המספר המינימלי עבור כל הקומות.

נציין כי $f(2,2)=2$ וגם $f(1,2)=1$.

במקרה הכללי:

$$f(n,b) \;=\; \min_{1\le i\le n}\; max(\,f(n-i,b)\;,\;f(i-1,b-1)\;)\;+1$$

הפונקציה $f(n,b)$ תהיה שווה ל- $1+log_2 n$ ($log_2 n$ בערך תחתון) כאשר $b\ge log_2 n$ (בערך עליון).



נציג מטריצת פתרון תכנות דינאמי עבור מקרה זה:

כמות ניסיונות		כדורים			
		0	**1**	**2**	**3**
קומות	**0**	0	0	0	0
	1	0	1	1	1
	2	0	2	2	2
	3	0	3	2	2
	4	0	4	3	3
	5	0	5	3	3
	6	0	6	3	3

הסבר למילוי המטריצה

עבור כל תא במטריצה נחשב את הנוסחה הבאה:

$$f(n,b) \;=\; \min_{1\le i\le n} \left[\, 1 + max(\, f(n-i,b) \,,\, f(i-1,b-1) \,) \,\right]$$

נזכר שכאשר זורקים כדור מקומה i יש שתי אפשרויות:

1. הכדור לא נשבר, נשארו b כדורים ו- $n-i$ קומות.
2. הכדור נשבר, נשאר b-1 כדורים ו- $i-1$ קומות.

הולכים על המקרה הגרוע ומחשבים את המספר המינימלי עבור כל הקומות.

למשל, עבור תא $(3,2)$ במטריצה שלנו, כלומר **3 קומות ו-2 כדורים**, נחשב בצורה הבאה:

אם אני זורק את הכדור הראשון מהקומה ה-i:	**אם הכדור לא נשבר** $f(n-i,b)$	**אם הכדור נשבר** $f(i-1,b-1)$	**1+ max**	$\min_{1\le i\le n}$
i = 1	$f(3-1,2)=f(2,2)=2$	$f(1-1,2-1)=f(0,1)=0$	$1+max(2,0)=3$	min[3,2,3] = **2**
i = 2	$f(3-2,2)=f(1,2)=1$	$f(2-1,2-1)=f(1,1)=1$	$1+max(1,1)=2$	
i = 3	$f(3-3,2)=f(0,2)=0$	$f(3-1,2-1)=f(2,1)=2$	$1+max(0,2)=3$	

לכן בתא (3,2) נציב $f(3,2)=2$**.**

למשל, עבור תא $(4,3)$ במטריצה שלנו, כלומר **4 קומות ו-3 כדורים**, נחשב בצורה הבאה:

$\min_{1\le i\le n}$	**1+ max**	**אם הכדור נשבר** $f(i-1,b-1)$	**אם הכדור לא נשבר** $f(n-i,b)$	**אם אני זורק את הכדור הראשון מהקומה ה-i:**
min[3,3,3,3] = **3**	$1+max(2,0)=3$	$f(1-1,3-1)=f(0,2)=0$	$f(4-1,3)=f(3,3)=2$	i = 1
	$1+max(2,1)=3$	$f(2-1,3-1)=f(1,2)=1$	$f(4-2,3)=f(2,3)=2$	i = 2
	$1+max(1,2)=3$	$f(3-1,3-1)=f(2,2)=2$	$f(4-3,3)=f(1,3)=1$	i = 3
	$1+max(0,2)=3$	$f(4-1,3-1)=f(3,2)=2$	$f(4-4,3)=f(0,3)=0$	i = 4

לכן בתא (4,3) נציב $f(4,3)=3$.

נמשיך להציב עד סוף המטריצה ונקבל כי עבור $f(6,3)=3$.

מימוש **אינדוקטיבי**

```
public static int minimalAttempts(int floors, int balls) {
   int[][] attempts = new int[floors+1][balls+1];
   for(int i = 0; i < attempts.length; i++) {
       attempts[i][0] = 0;
       attempts[i][1] = i;
   }
   for(int j = 1 ; j < attempts[0].length; j++) {
       attempts[0][j] = 0;
       attempts[1][j] = 1;
   }
   for(int b = 2; b < attempts[0].length; b++) { // balls
       for(int n = 2; n < attempts.length ; n++) { // floors
           int min = Integer.MAX_VALUE;
           for(int i = 1; i <= n; i++) {
               int max = Math.max(attempts[n-i][b], attempts[i-1][b]) + 1;
               if(min > max) { min = max; }
           }
           attempts[n][b] = min;
       }
   }

   return attempts[floors][balls];
}

public static void main(String[] args) {
   System.out.println("minimal: " + minimalAttempts(6,3));
}
```

מימוש **רקורסיבי**

```java
public static int minimalAttempts(int floors, int balls) {
   int[][] attempts = new int[floors+1][balls+1];
   for(int i = 0; i < attempts.length; i++) {
       attempts[i][0] = 0;
       attempts[i][1] = i;
   }
   for(int j = 1 ; j < attempts[0].length; j++) {
       attempts[0][j] = 0;
       attempts[1][j] = 1;
   }
   generateRec(attempts,2,2,1, Integer.MAX_VALUE);
   printRec(attempts,0,0);
   return attempts[floors][balls];
}
private static void generateRec(int[][] attempts, int b, int n, int i, int min) {
   if(b == attempts[0].length) {
       return;
   } else if(n == attempts.length) {
       generateRec(attempts,b+1,2,1,min);
   } else if(i == n) {
       attempts[n][b] = min;
       generateRec(attempts,b,n+1,1,Integer.MAX_VALUE);
   } else {
       int max = Math.max(attempts[n-i][b], attempts[i-1][b]) + 1;
       if(min > max) {
           min = max;
       }
       generateRec(attempts,b,n,i+1,min);
   }
}
public static void printRec(int[][] attempts, int i, int j) {
   if(i == attempts.length) {
       return;
   } else if(j == attempts[0].length) {
       System.out.println();
       printRec(attempts,i+1,0);
   } else {
       System.out.print(attempts[i][j] + "\t");
       printRec(attempts,i,j+1);
   }
}
public static void main(String[] args) {
   System.out.println("minimal: " + minimalAttempts(6,3));
}
```

סיבוכיות זמן ריצה (לא כולל הדפסה) : $O(n^2 \cdot b)$. n=מס' הקומות , b = מספר הכדורים.