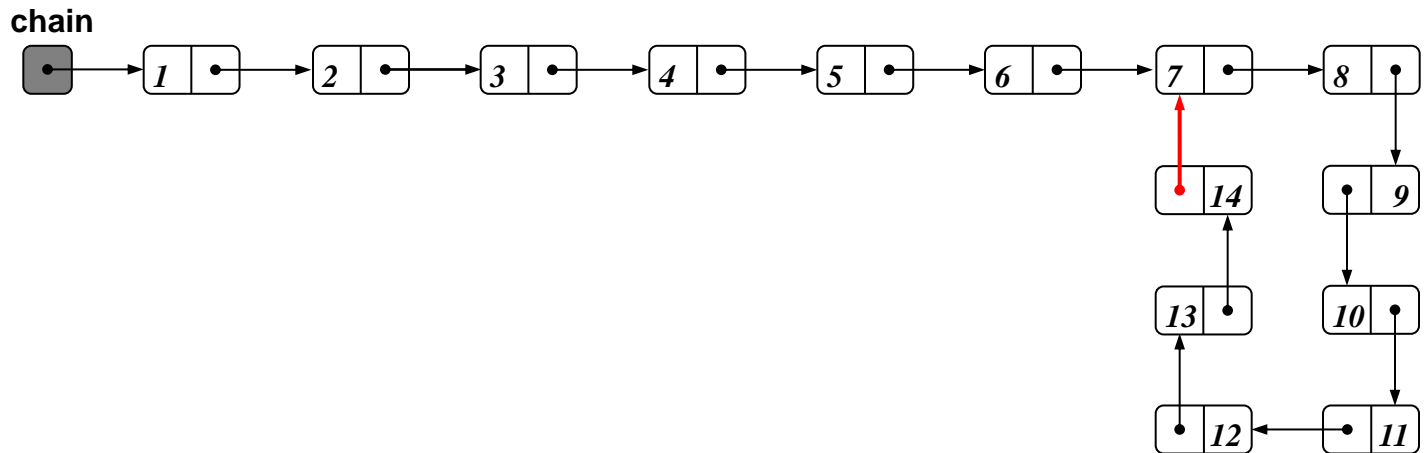


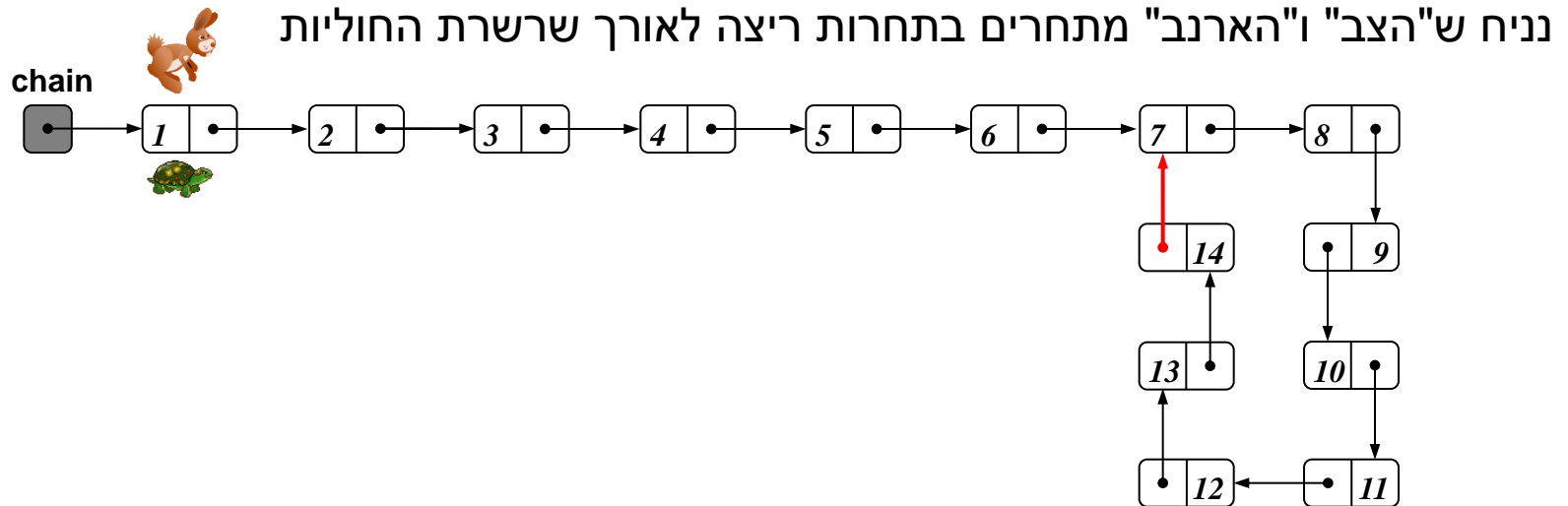
אלגוריתם "הארנב והצב"



האלגוריתם הוצג לראשונה על ידי המדען האמריקאי פלוייד (Robert W. Floyd). האלגוריתם גם ידוע בשם "הארנב והצב".

שם האלגוריתם "הארנב והצב" נקרא על שם המְשֶׁל של איזופוס. המשל מספר על תחרות ריצה המתקיימת בין צב לבין ארנב.

אלגוריתם "הארנב והצב"

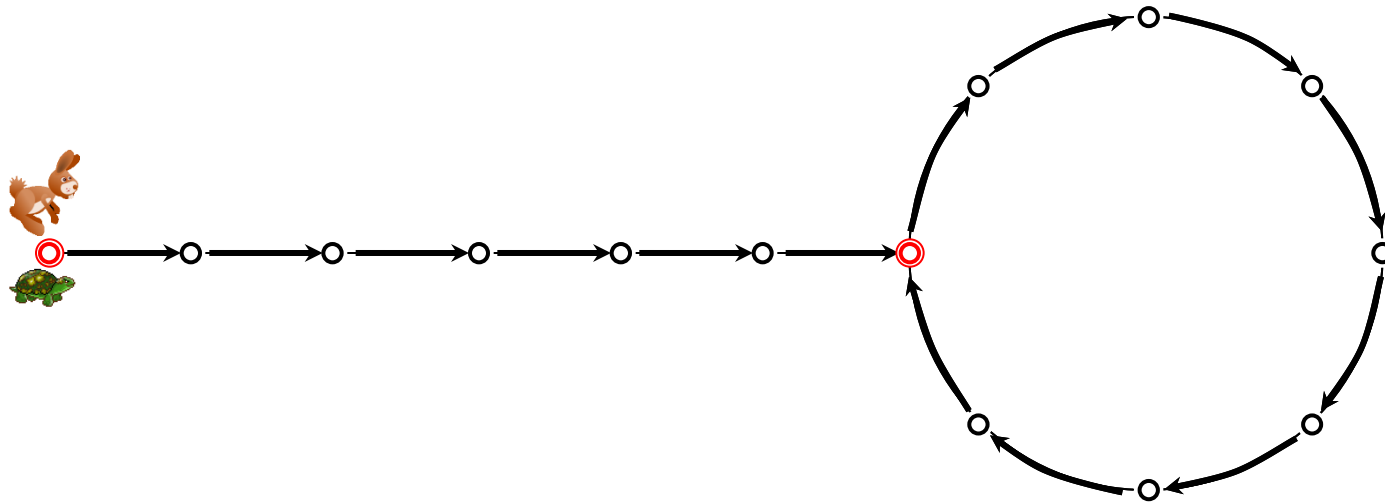


כל אחד מהמתחרים רץ בקצב שלו:

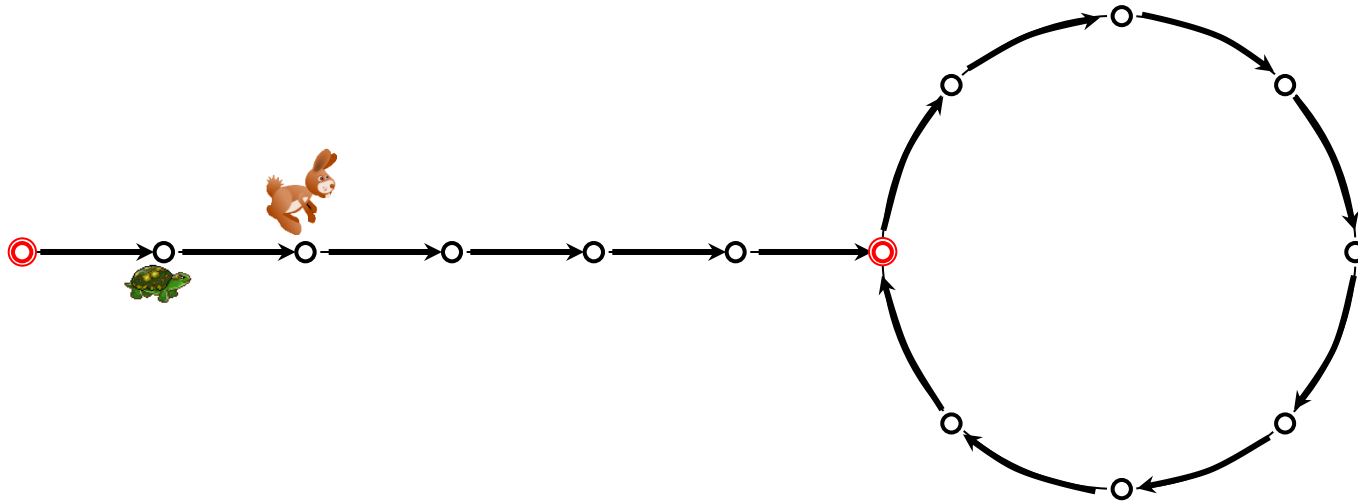
- "הצב" רץ בקצב איטי חוליה אחרי חוליה
- "הארנב" רץ בקפיצות של שתי חוליות

הטענה היא שאם השרשרת היא מעגלית - המתחרים ייפגשו שוב בחוליה מסוימת על השרשרת. לעומת זאת, אם השרשרת אינה מעגלית המתחרים לא ייפגשו, ולמעשה "הארנב" יגיע קודם לסוף השרשרת.

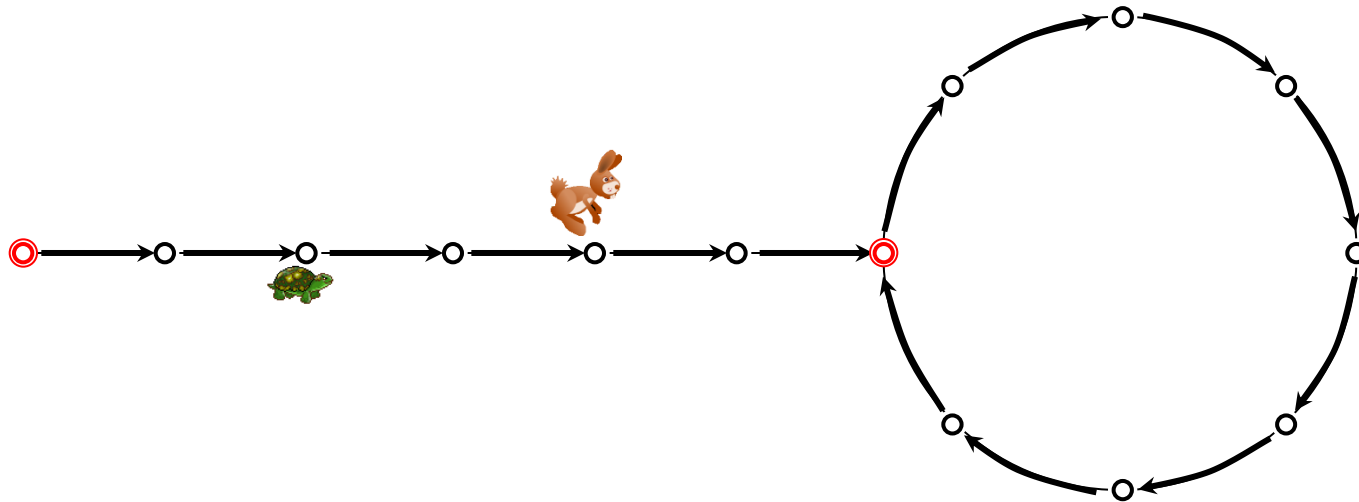
אלגוריתם "הארנב והצב"



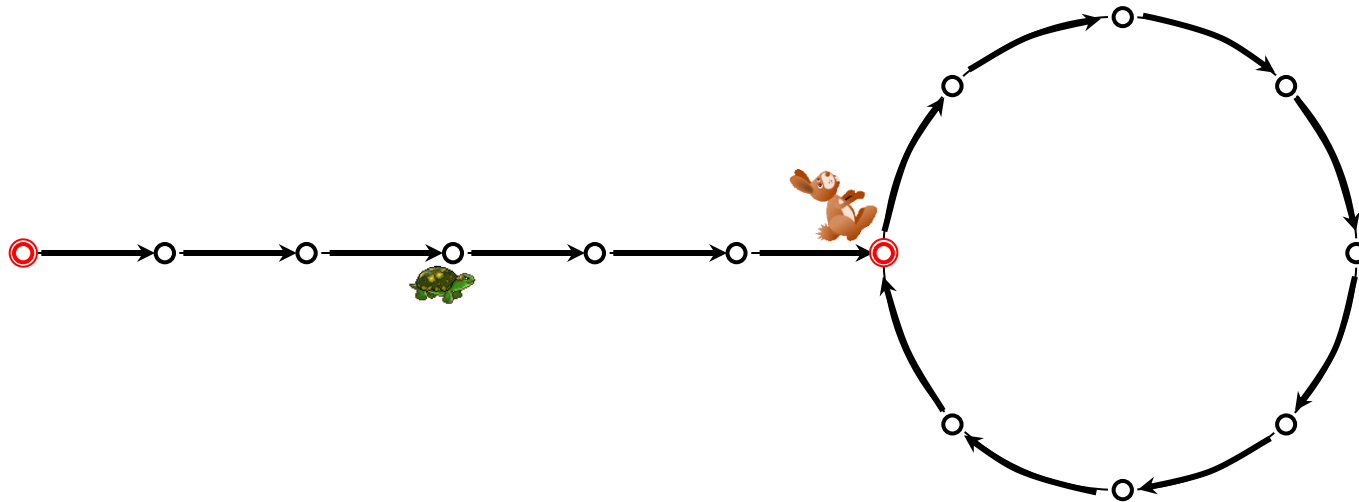
אלגוריתם "הארנב והצב"



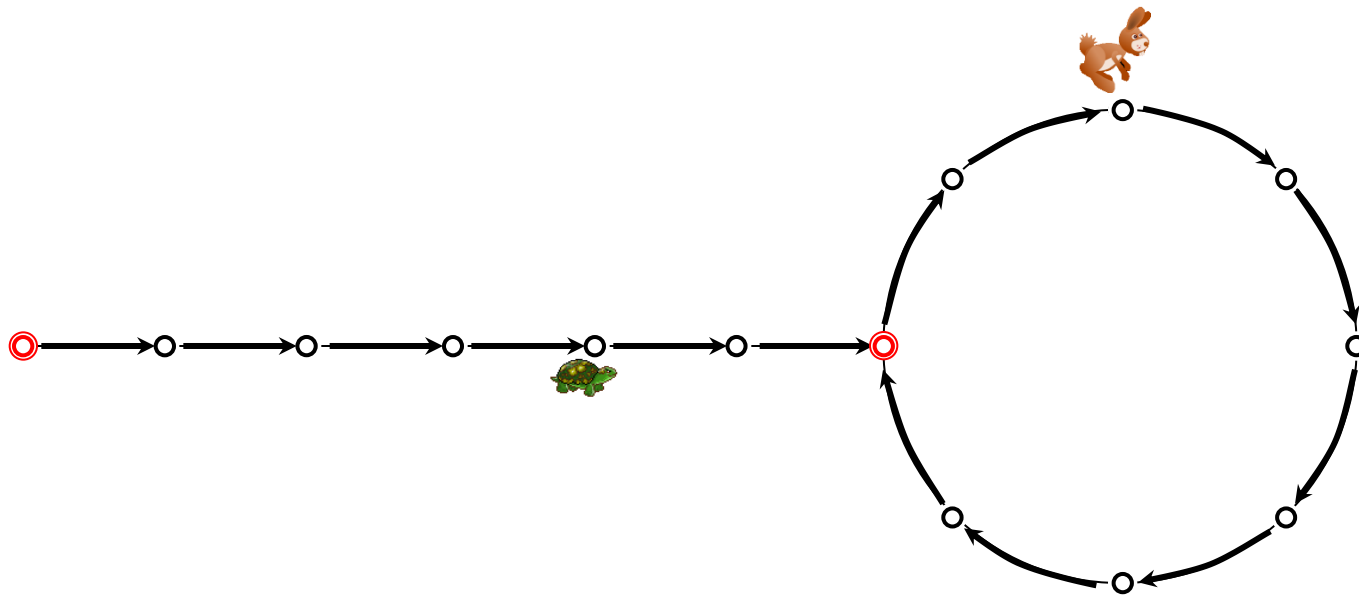
אלגוריתם "הארנב והצב"



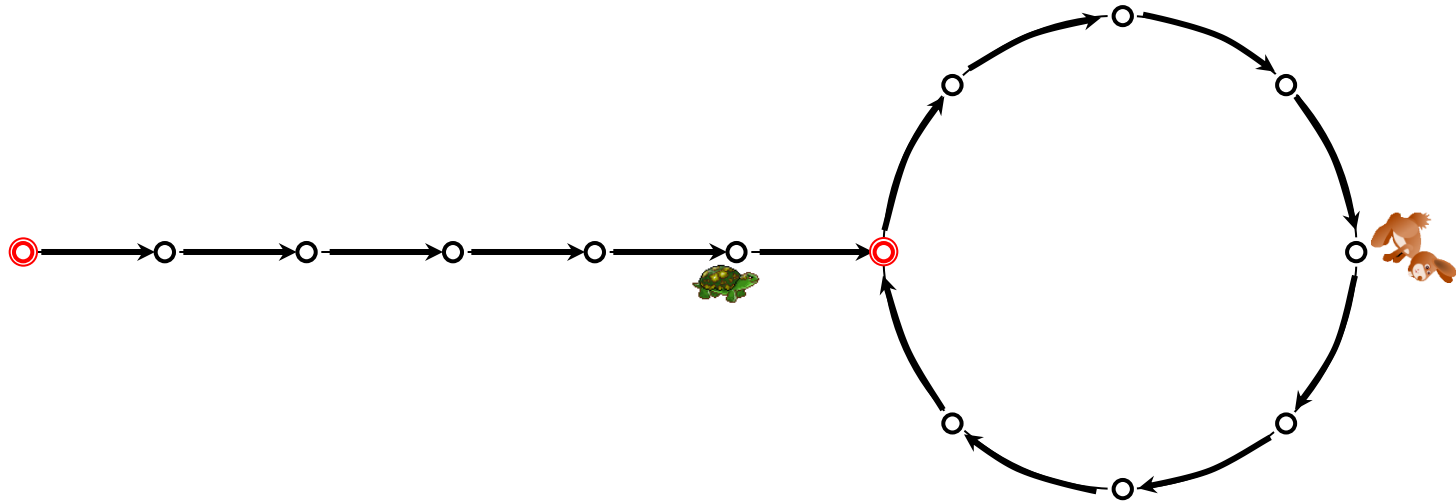
אלגוריתם "הארנב והצב"



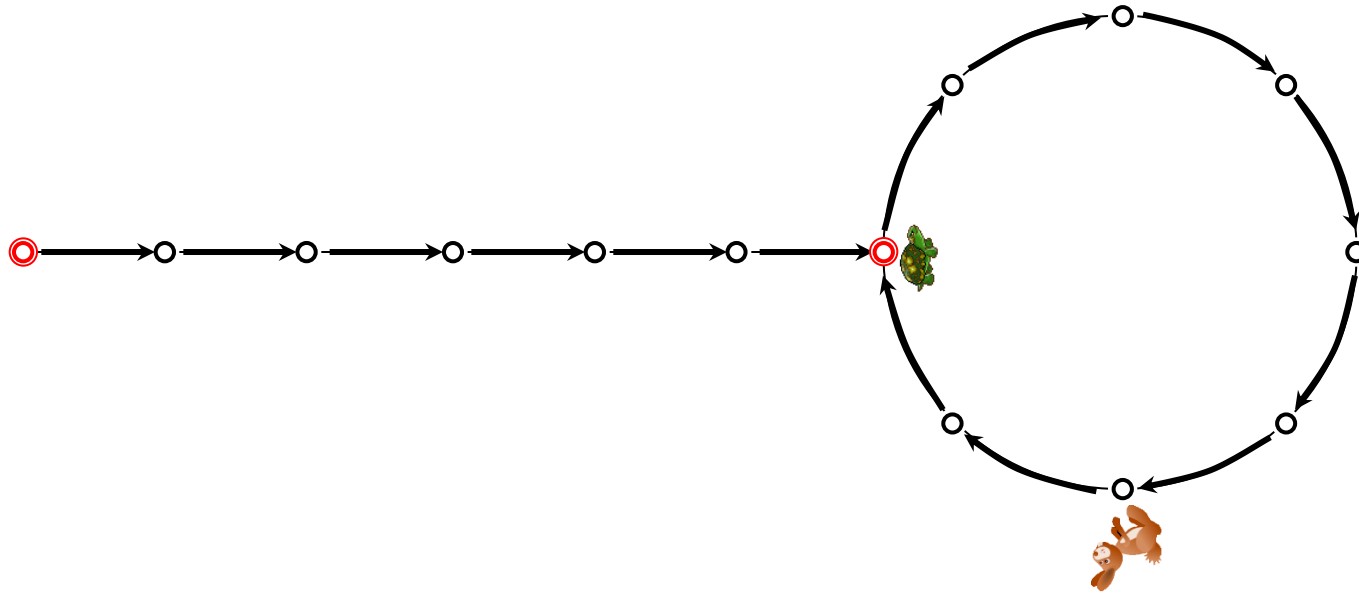
אלגוריתם "הארנב והצב"



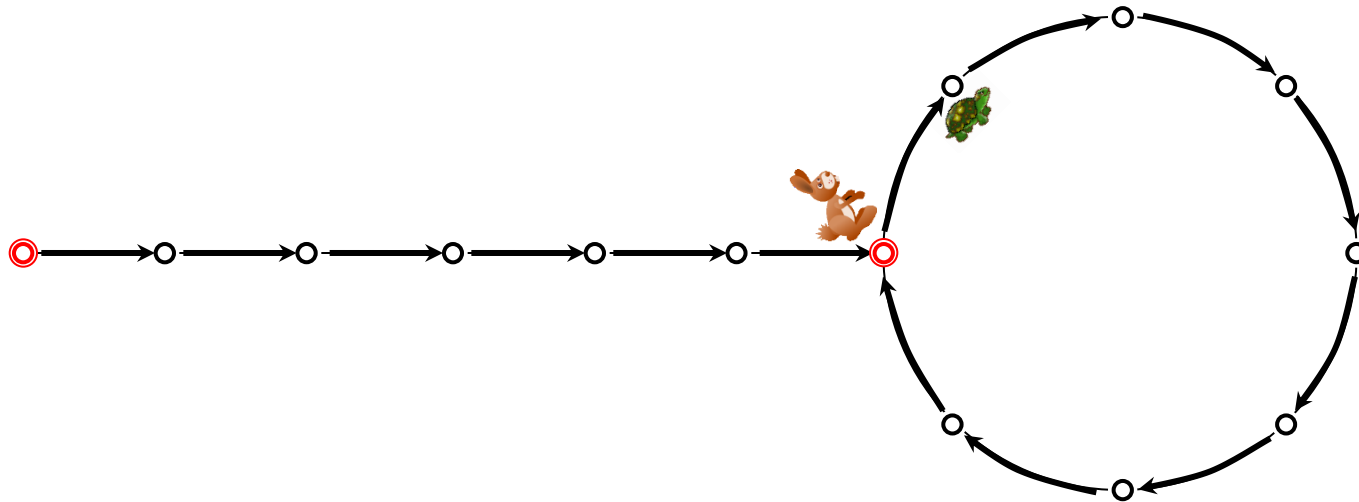
אלגוריתם "הארנב והצב"



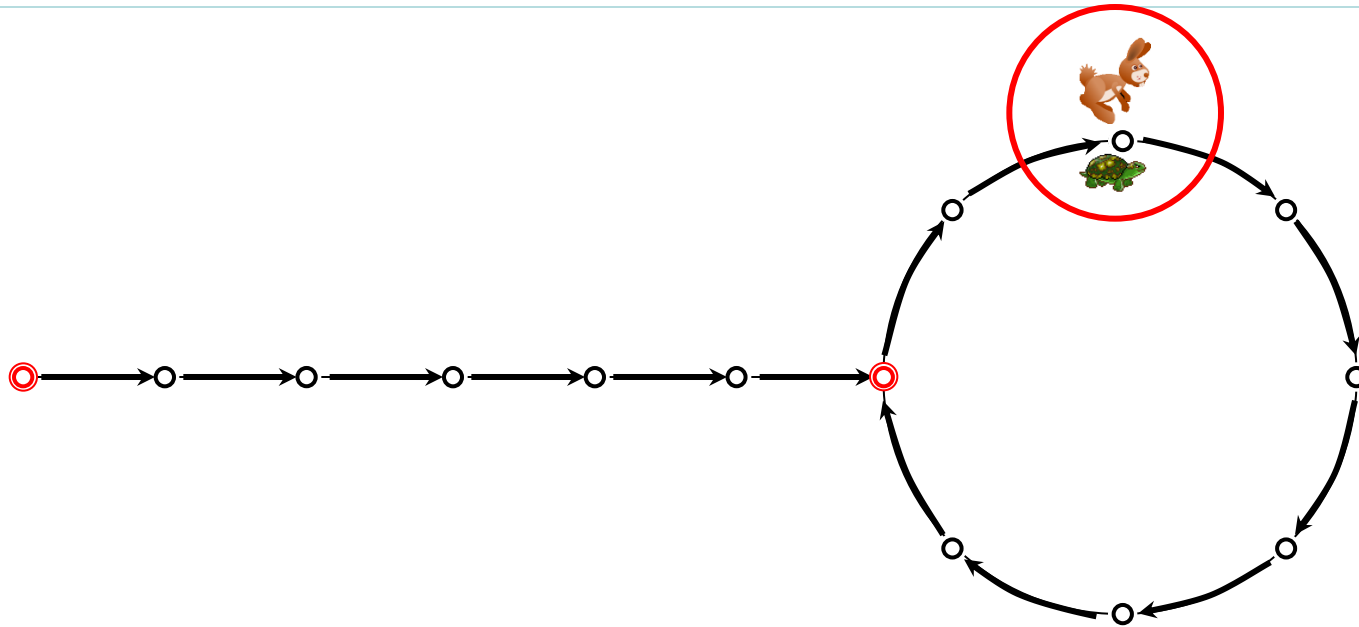
אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"

כתבו פעולה המקבלת שרשרת חוליות ומחזיר **true** אם היא שרשרת מעגלית ו-**false** אם היא לינארית (שרשרת שבה החוליה האחרונה מצביע ל-**null**)

אלגוריתם "הארנב והצב"

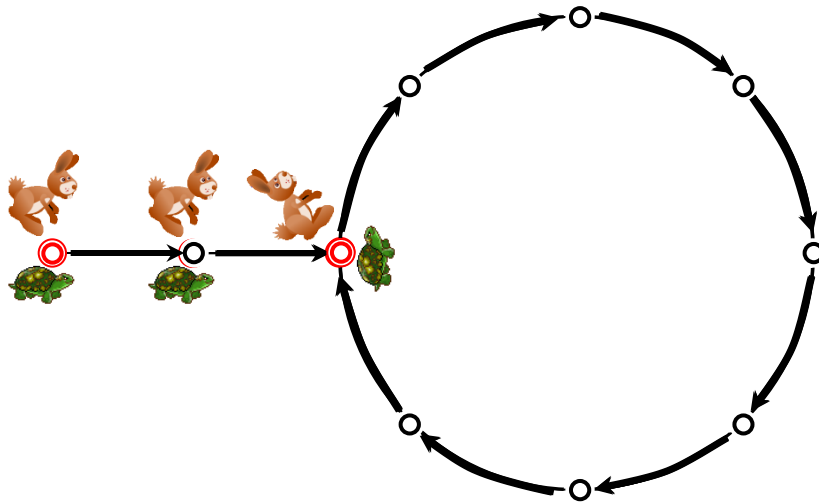
```
public static <T> boolean isCircled(Node<T> chain){
    Node<T> fast = chain;
    Node<T> slow = chain;
    do {
        slow = slow.getNext();
        fast = fast.getNext();
        if (fast == null)
            return false;
        fast = fast.getNext();
        if (fast == null)
            return false;
    } while (fast != slow)
    return true;
}
```

אלגוריתם "הארנב והצב"

שאלת אתגר

כתבו פעולה המקבלת שרשרת חוליות מעגלית או לינארית, מחפשת ומחזירה הפניה לחוליה בה מתחיל המעגל, ואם השרשרת אינה מעגלית הפעולה מחזירה `null`. יש לכתוב את הפעולה בסיבוכיות לינארית $O(n)$, וללא שימוש בזיכרון נוסף.

ניתן להגיע לפתרון הבעיה, באופן עצמאי, מחקירת המקרים הבאים:

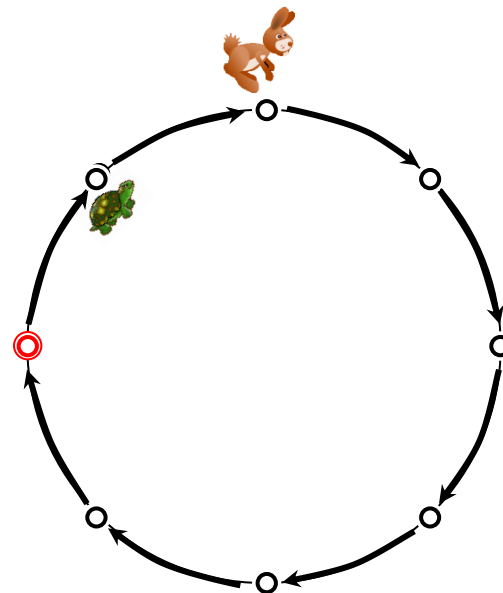


1. שרשרת חוליות מעגלית בלי "זנב".
2. שרשרת חוליות מעגלית עם "זנב" באורך 1.
3. שרשרת חוליות מעגלית עם "זנב" באורך 2.
4. וכן הלאה ...

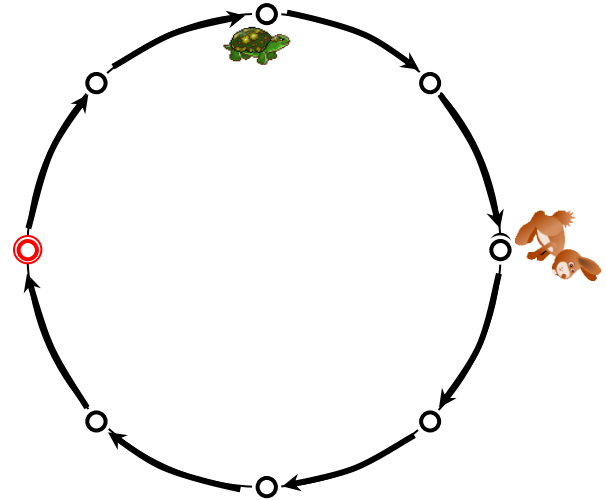
אלגוריתם "הארנב והצב"

שרשרת חוליות מעגלית בלי "זנב"

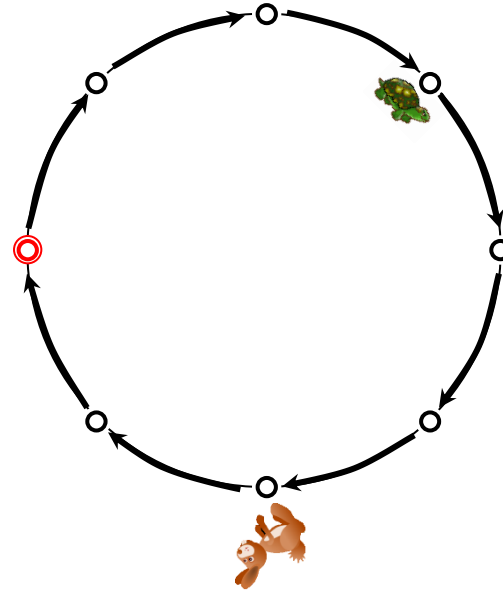
אלגוריתם "הארנב והצב"



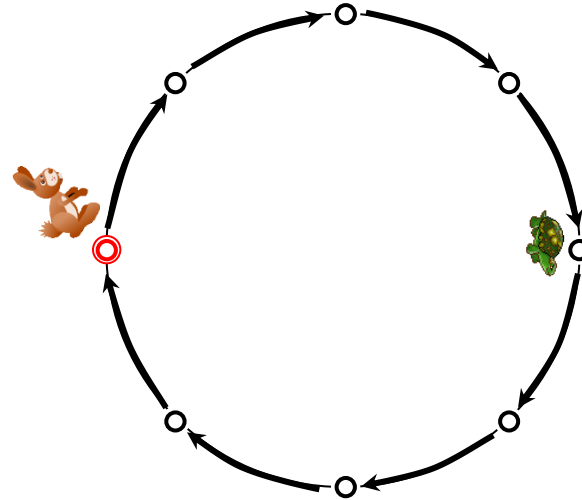
אלגוריתם "הארנב והצב"



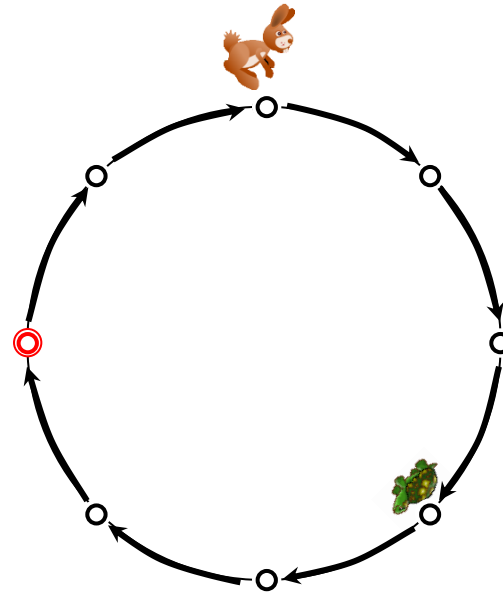
אלגוריתם "הארנב והצב"



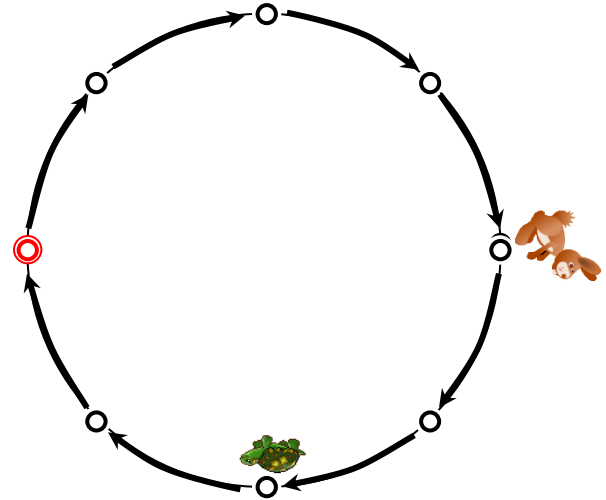
אלגוריתם "הארנב והצב"



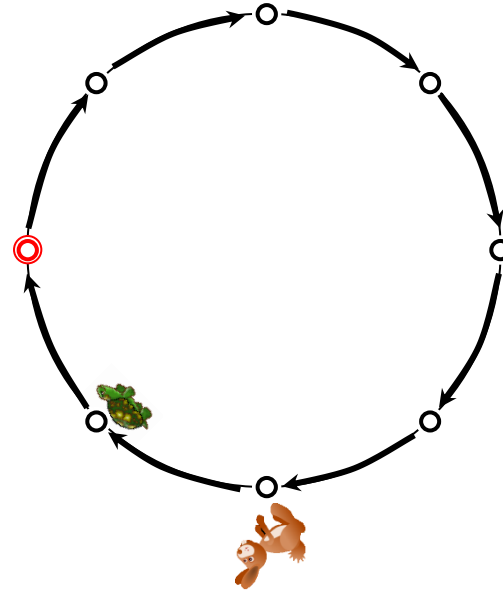
אלגוריתם "הארנב והצב"



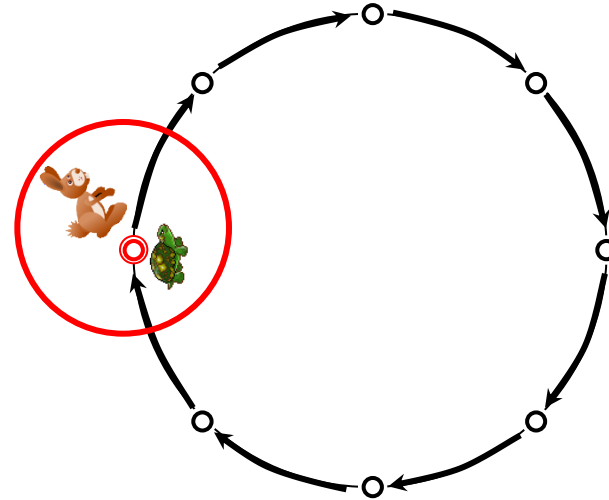
אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"

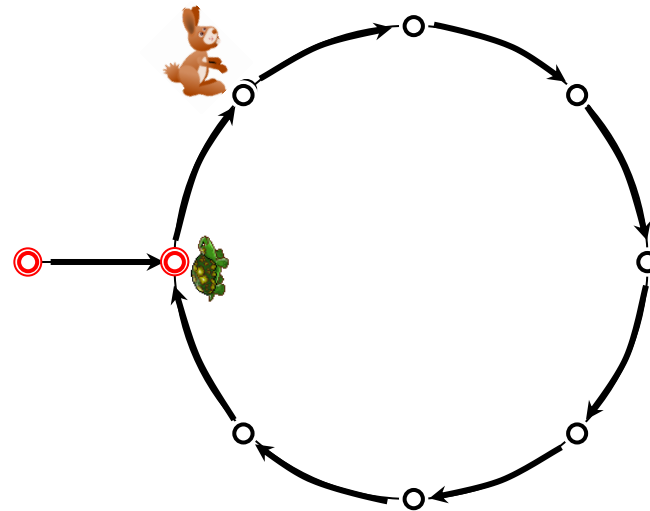


בשרשרת החוליות המעגלית בלי "זנב" הארנב והצב ייפגשו שוב בתחילת השרשרת.

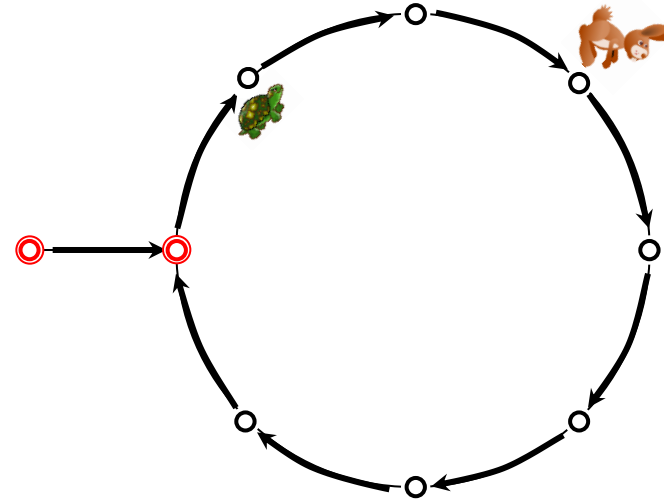
אלגוריתם "הארנב והצב"

שרשרת חוליות חלקית מעגלית עם "זנב" באורך 1

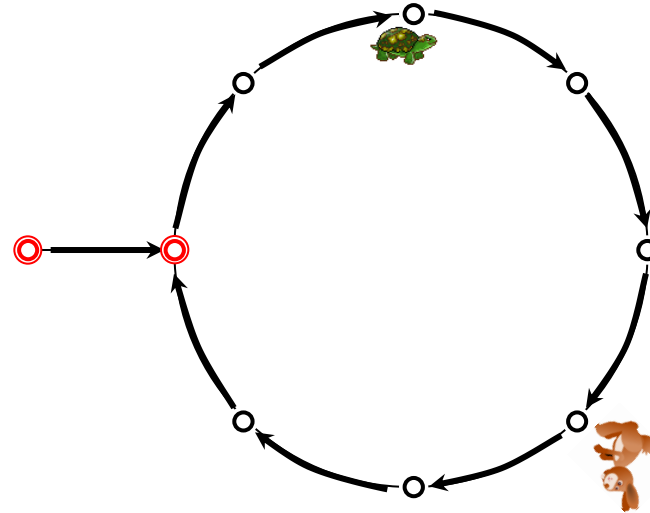
אלגוריתם "הארנב והצב"



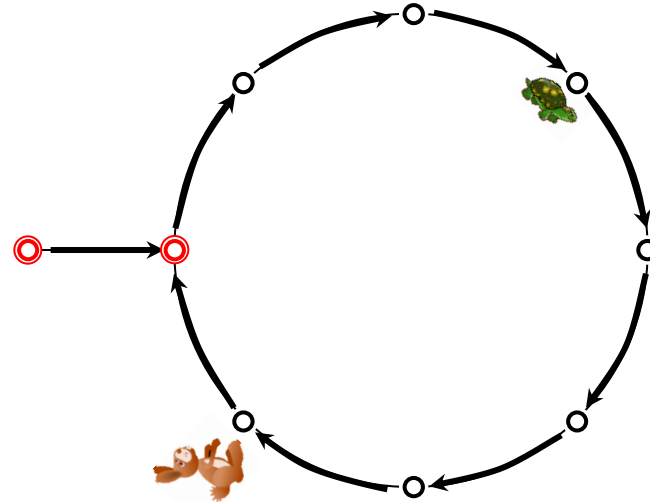
אלגוריתם "הארנב והצב"



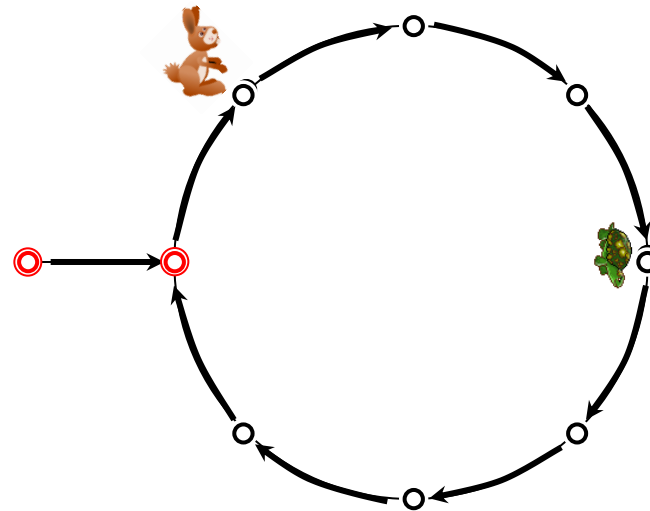
אלגוריתם "הארנב והצב"



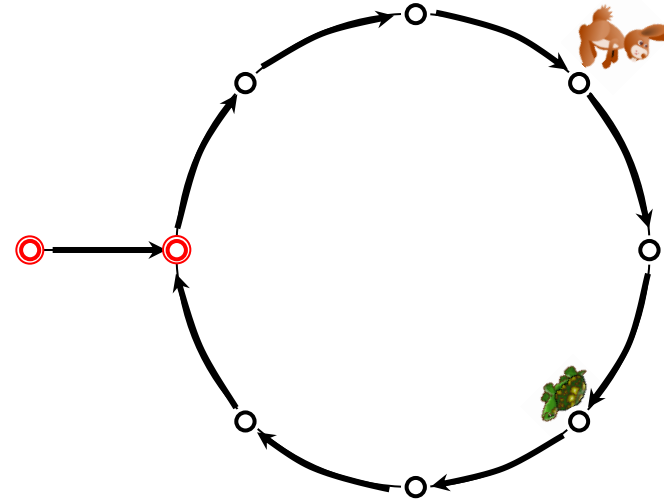
אלגוריתם "הארנב והצב"



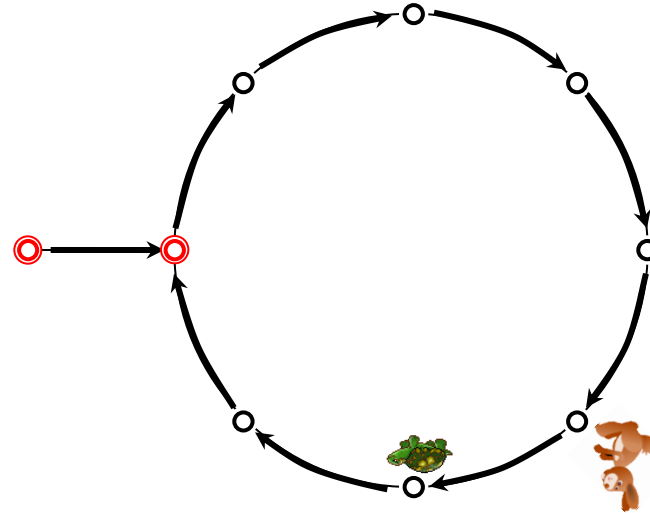
אלגוריתם "הארנב והצב"



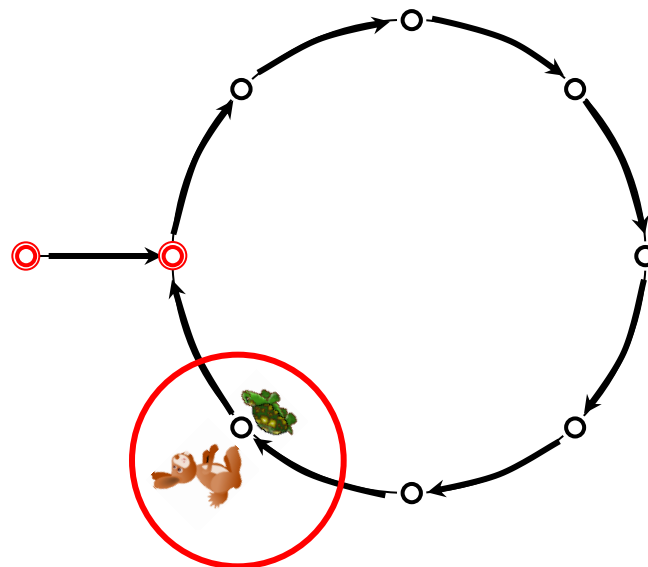
אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"

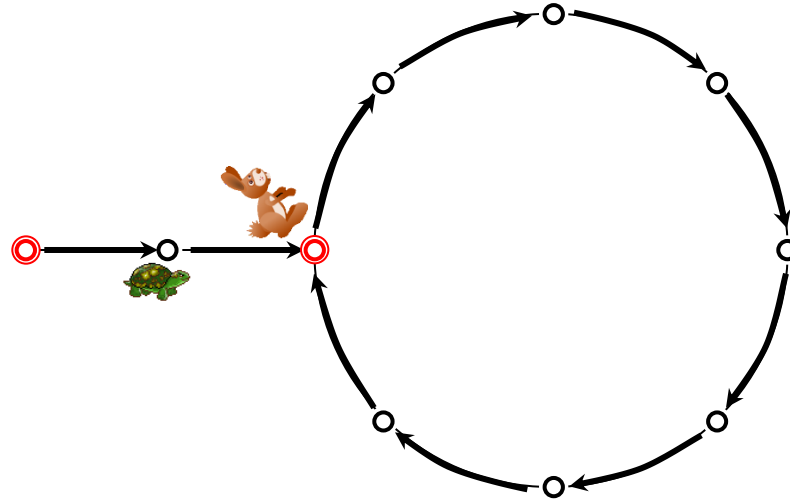


בשרשרת החוליות המעגלית עם "זנב" באורך 1 הארנב והצב ייפגשו שוב בשלב אחד לפני תחילת המעגל.

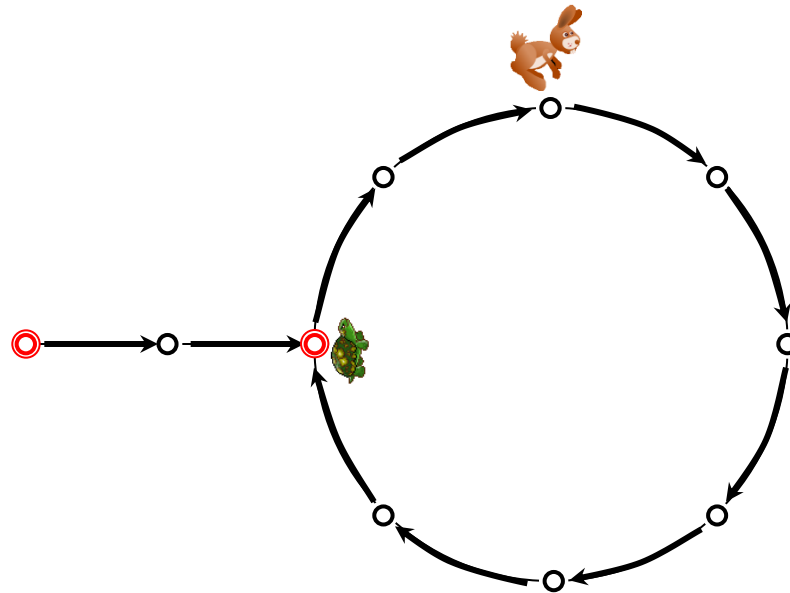
אלגוריתם "הארנב והצב"

שרשרת חוליות חלקית מעגלית עם "זנב" באורך 2

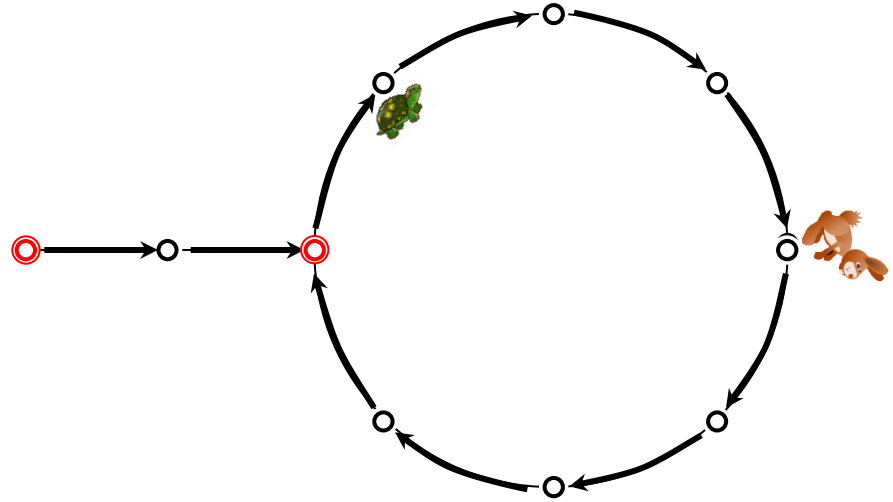
אלגוריתם "הארנב והצב"



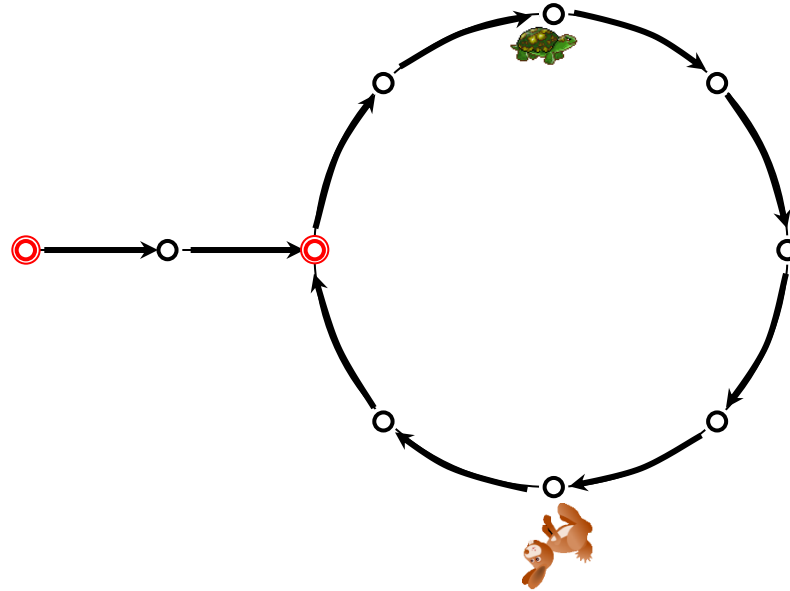
אלגוריתם "הארנב והצב"



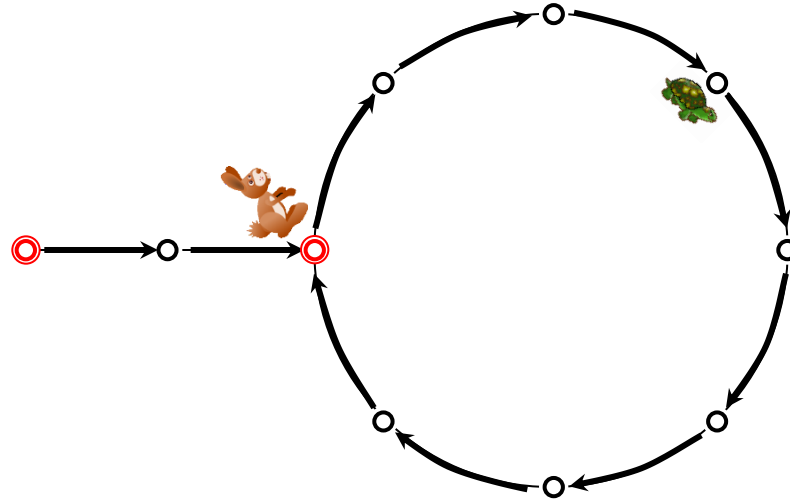
אלגוריתם "הארנב והצב"



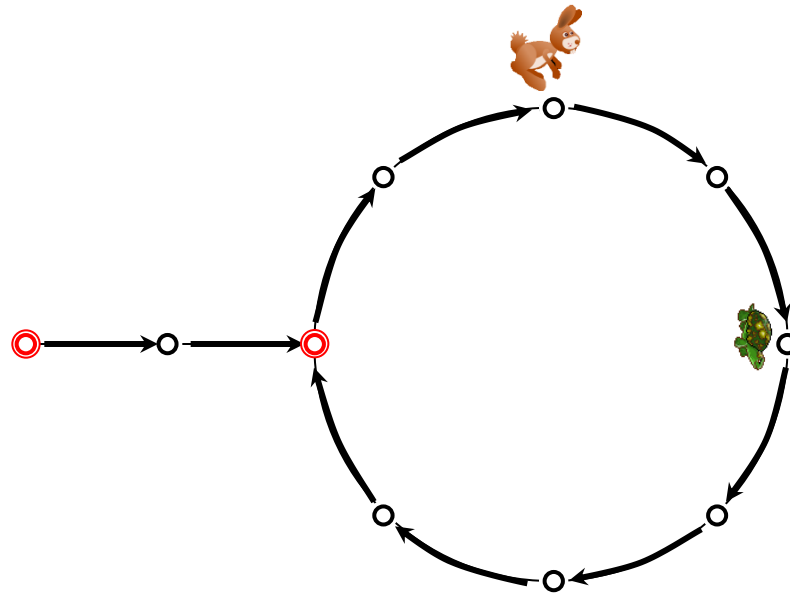
אלגוריתם "הארנב והצב"



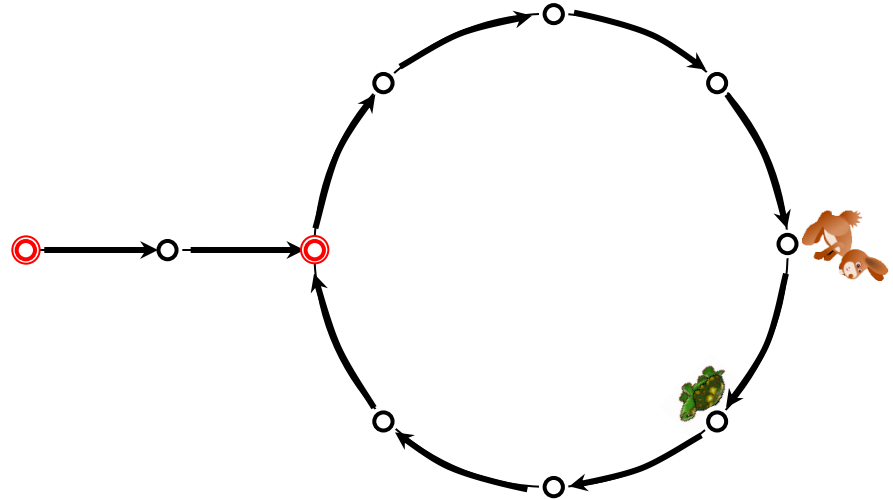
אלגוריתם "הארנב והצב"



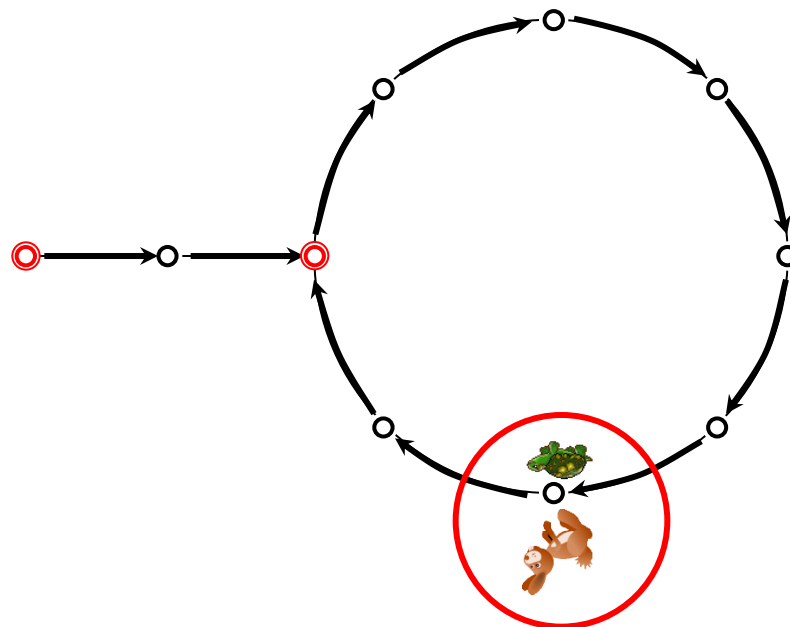
אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"



אלגוריתם "הארנב והצב"



בשרשרת החוליות המעגלית עם "זנב" באורך 2 הארנב והצב ייפגשו שוב בשני שלבים לפני תחילת המעגל.

אלגוריתם "הארנב והצב"

מסקנות:

- בשרשרת חוליות מעגלית בלי "זנב" הארנב והצב ייפגשו בתחילת המעגל. זה גם יקרה אם שארית בחלוקה של אורך "זנב" באורך המעגל שווה לאפס.
$$\text{אורך המעגל} \% \text{אורך "זנב"} = 0$$

- בשרשרת חוליות מעגלית עם "זנב" באורך 1 הארנב והצב ייפגשו בשלב אחד לפני תחילת המעגל. זה גם יקרה אם שארית בחלוקה של אורך "זנב" באורך המעגל שווה ל-1.
$$\text{אורך המעגל} \% \text{אורך "זנב"} = 1$$

- בשרשרת חוליות מעגלית עם "זנב" באורך 2 הארנב והצב ייפגשו בשני שלבים לפני תחילת המעגל. זה גם יקרה אם שארית החלוקה של אורך "זנב" באורך המעגל שווה ל-2.
$$\text{אורך המעגל} \% \text{אורך "זנב"} = 2$$

- וכן הלא.

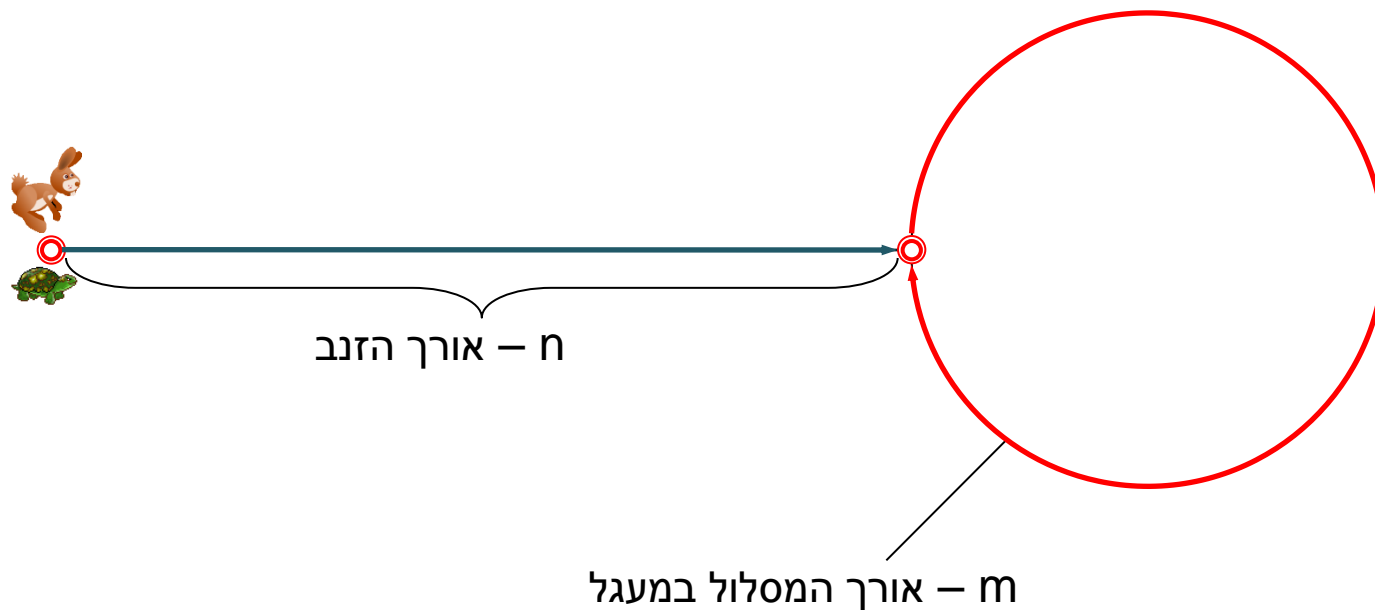
פתרון:

נחזיק הפניה אחת לחוליה הראשונה בשרשרת, והפניה נוספת לחוליה במקום המפגש בין הארנב והצב. כעת נזיז במקביל את שתי ההפניות לאורך השרשרת חוליה אחרי חוליה (כמו "הצב"). בכל שלב, נבדוק האם הן נפגשות.

אלגוריתם "הארנב והצב"

הוכחה:

נסמן

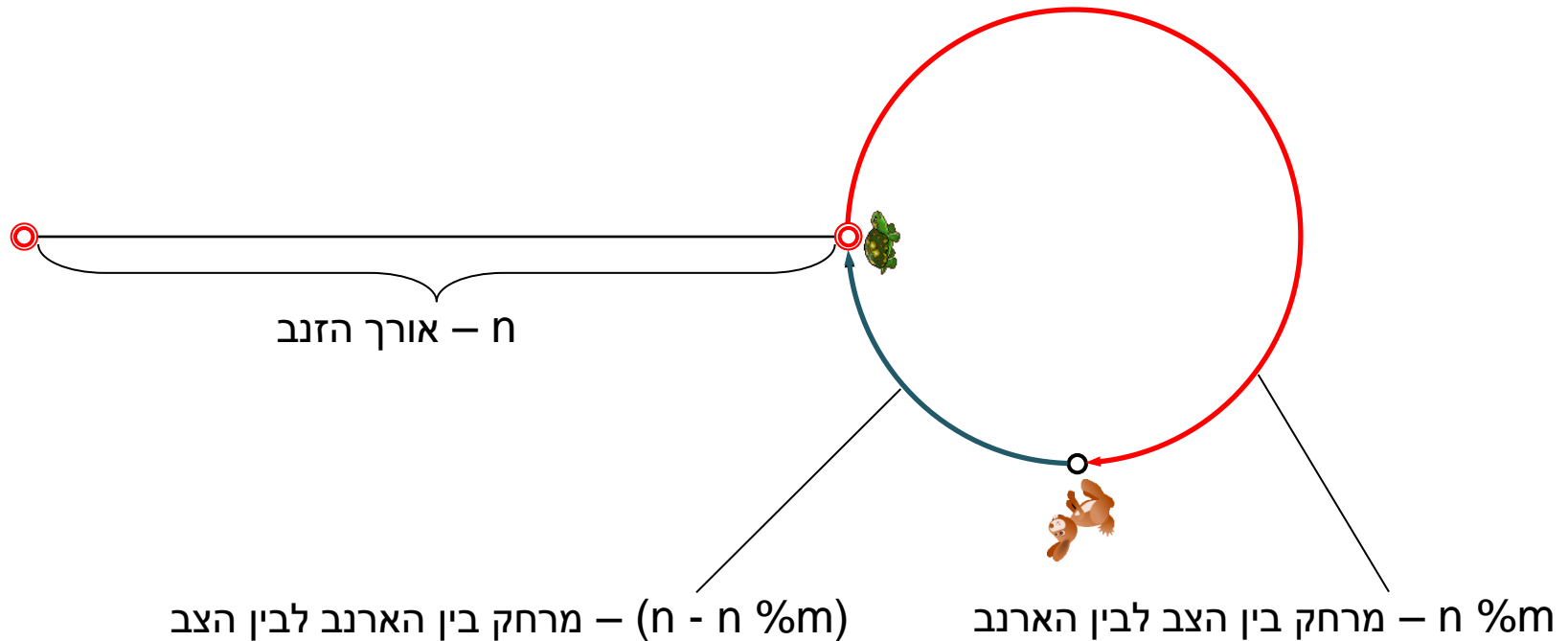


מה יהיה המרחק בין הצב לבין הארנב, כאשר הצב מגיע לתחילת המעגל?

אלגוריתם "הארנב והצב"

הוכחה:

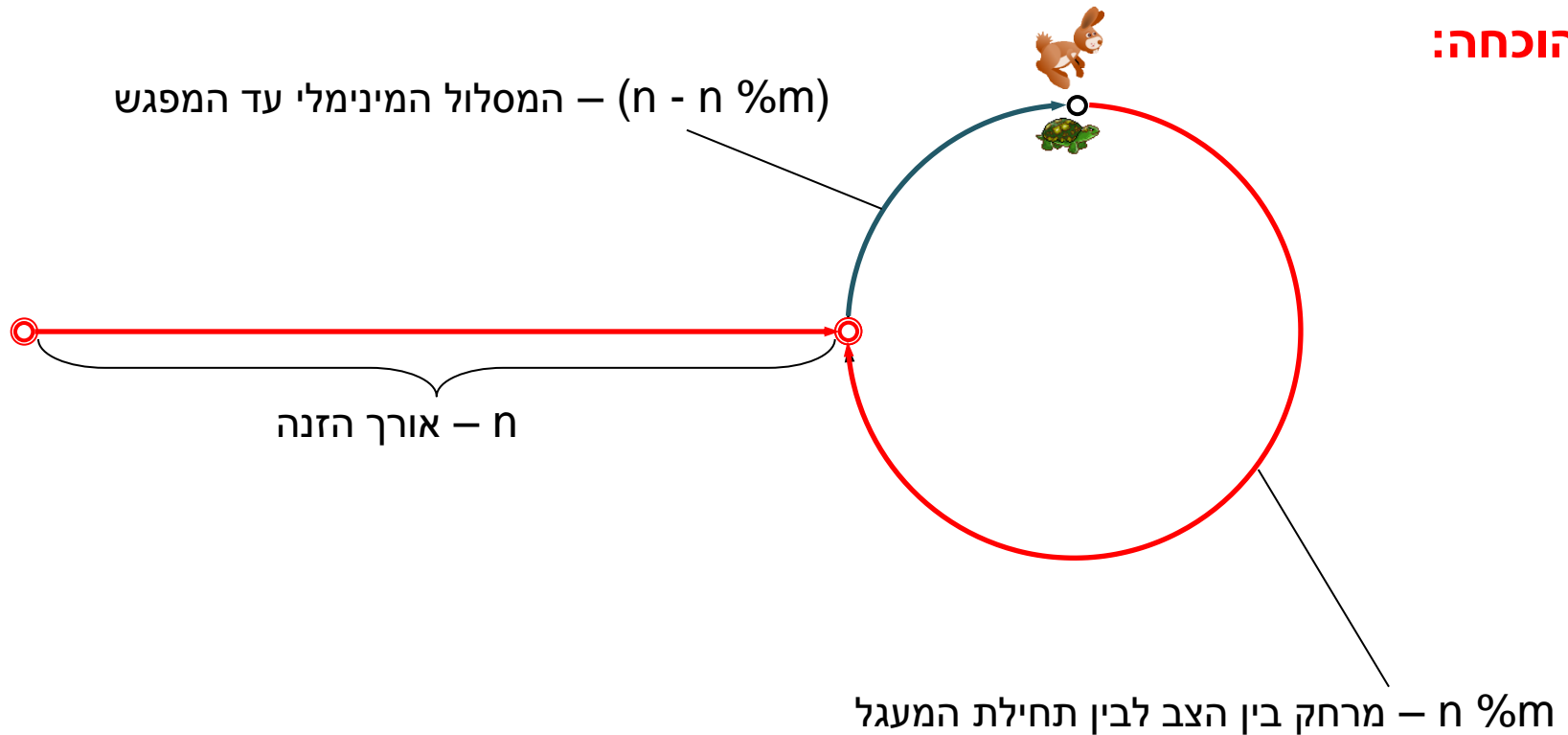
כאשר הצב מגיע לתחילת המעגל



מהו אורך המסלול המינימלי שעובר את הצב עד למפגש עם הארנב במעגל?

אלגוריתם "הארנב והצב"

הוכחה:

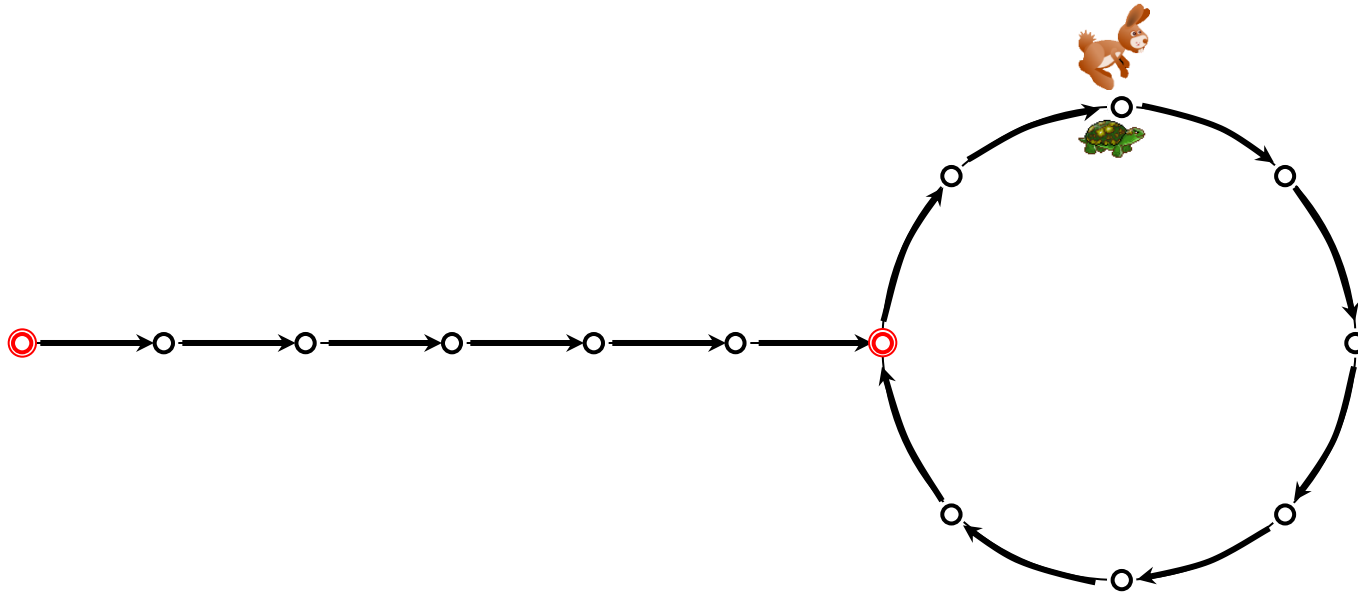


אורך הזנב ומרחק בין הצב לבין תחילת המעגל שקולים מודולו m

$$n \equiv n \% m$$

אלגוריתם "הארנב והצב"

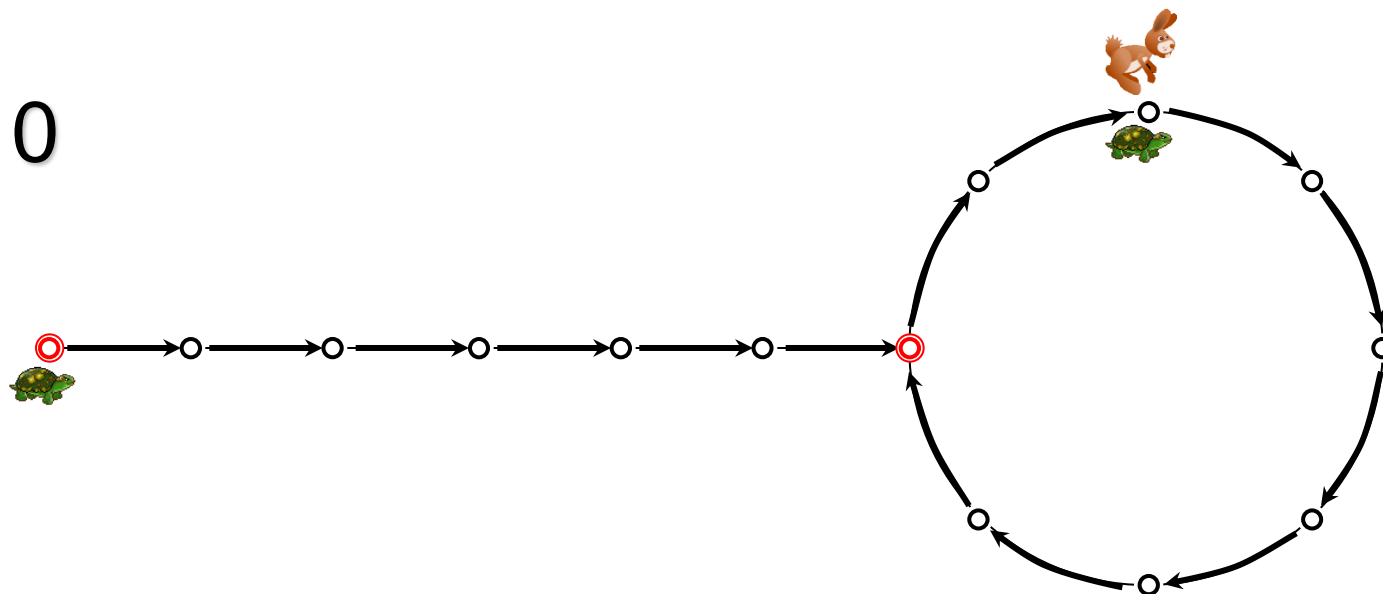
הדמיה



נשים הפניה בחוליה הראשונה בשרשרת.

אלגוריתם "הארנב והצב"

הדמיה

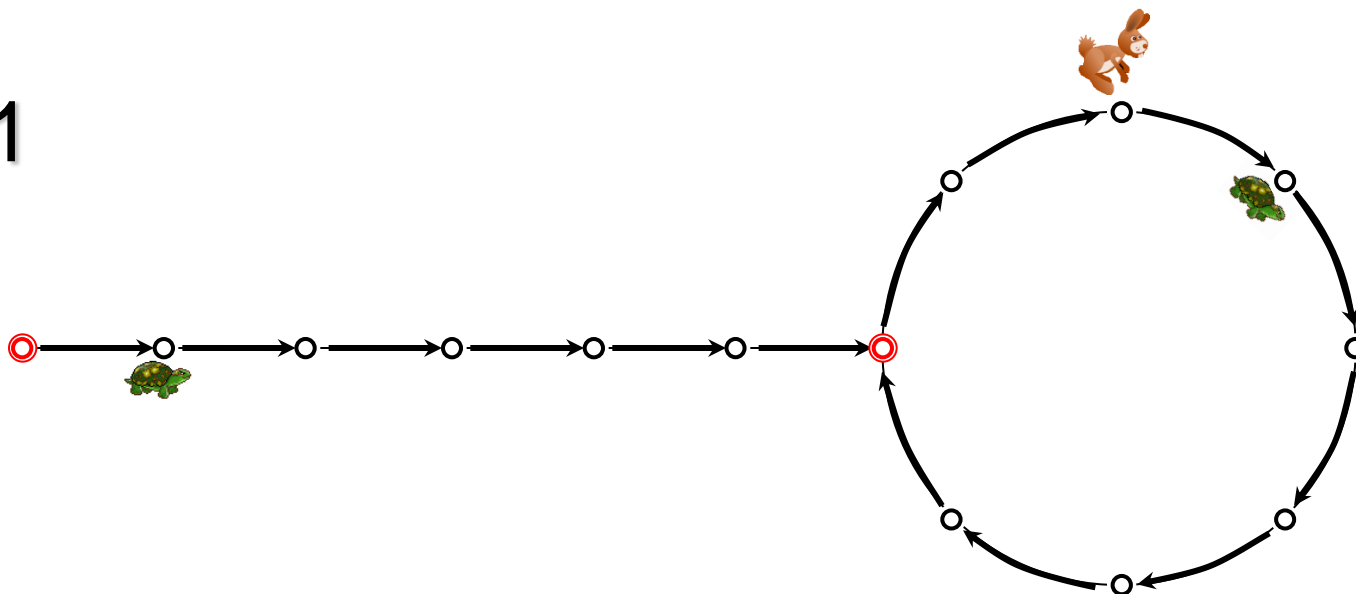


כעת נזיז במקביל את שתי ההפניות לאורך השרשרת חוליה אחרי חוליה (כמו "הצב").

אלגוריתם "הארנב והצב"

הדמיה

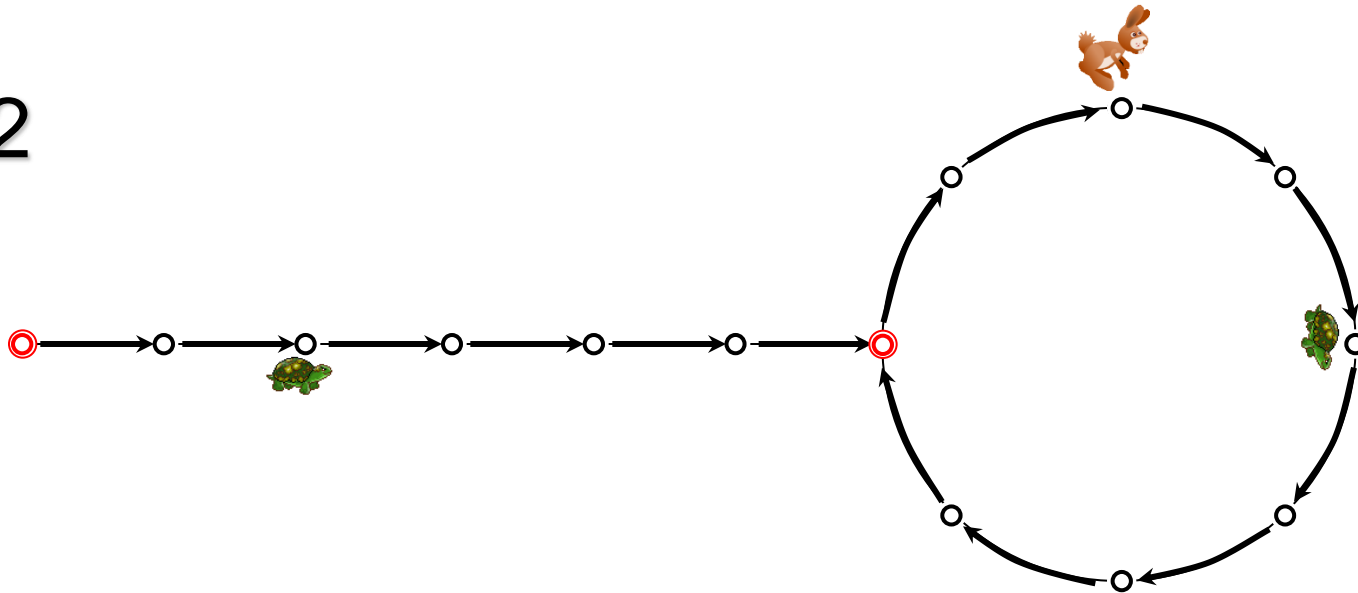
1



אלגוריתם "הארנב והצב"

הדמיה

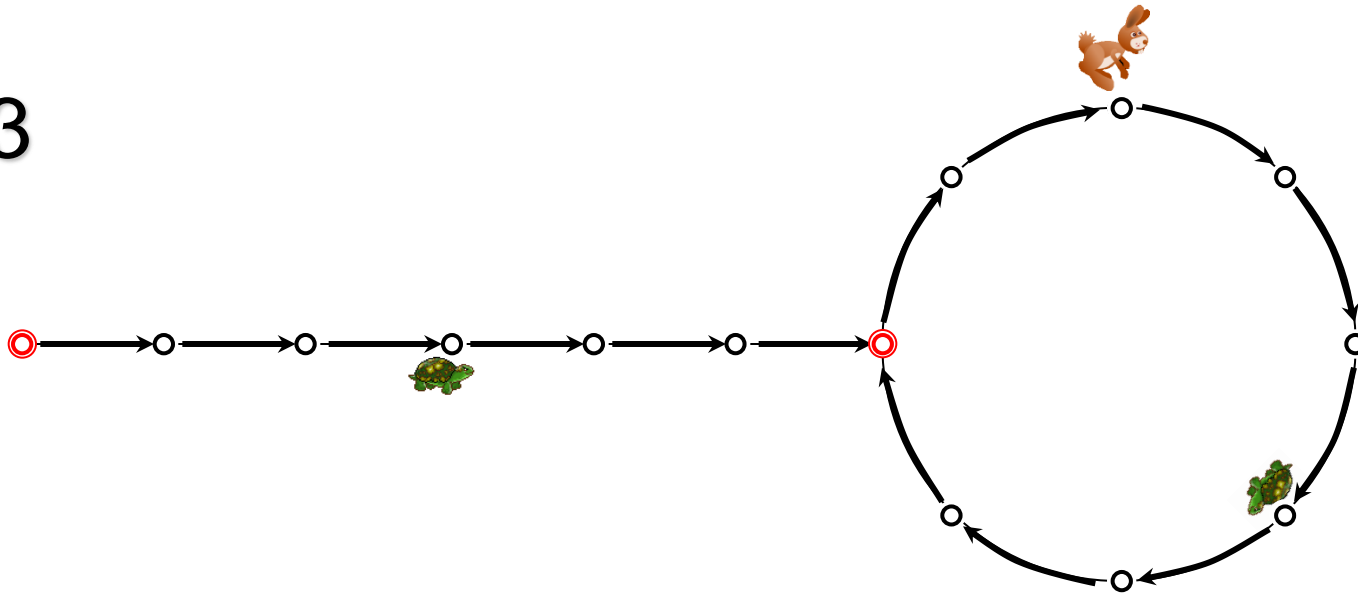
2



אלגוריתם "הארנב והצב"

הדמיה

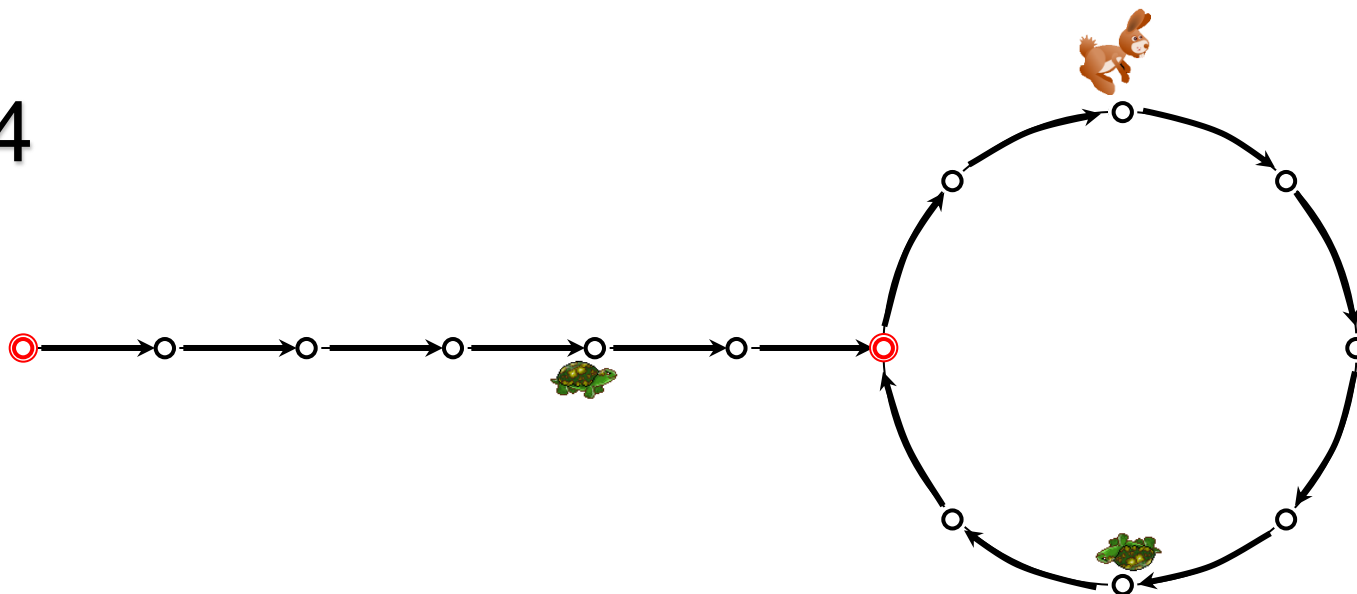
3



אלגוריתם "הארנב והצב"

הדמיה

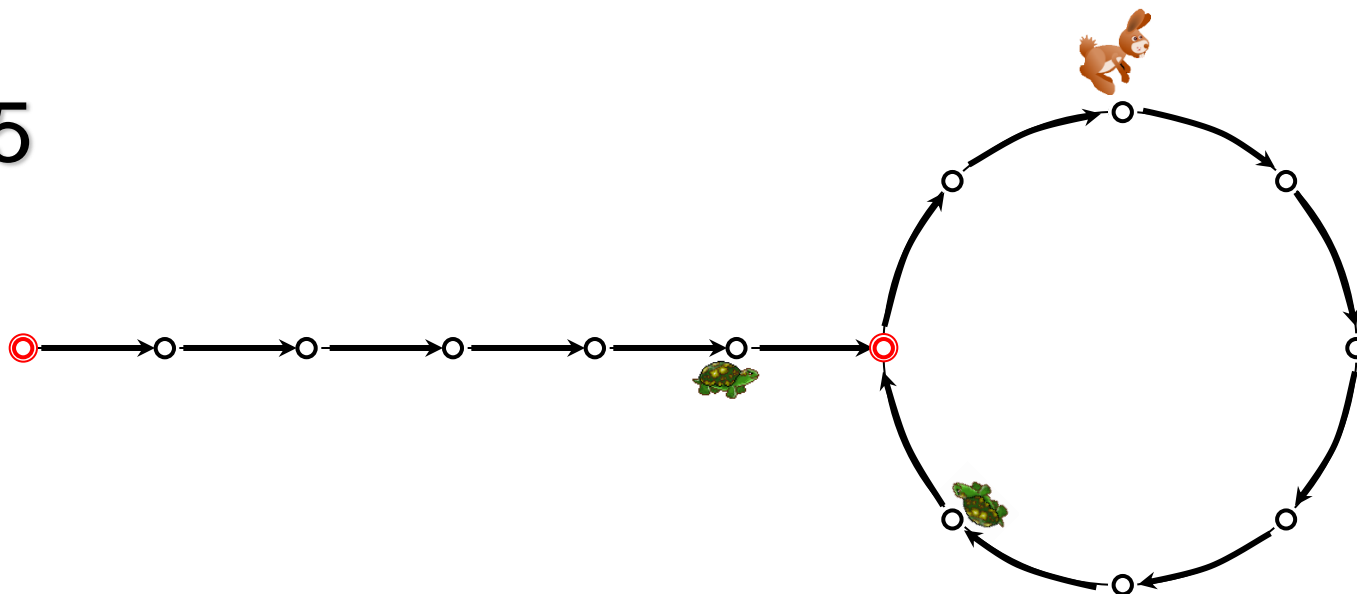
4



אלגוריתם "הארנב והצב"

הדמיה

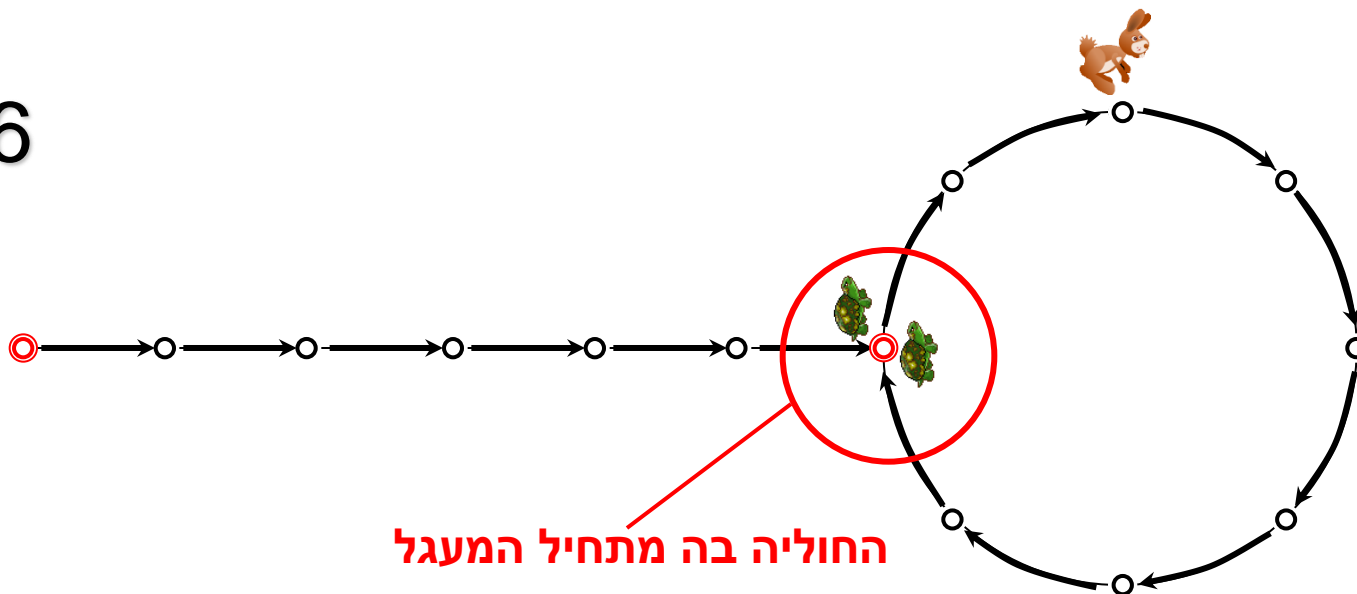
5



אלגוריתם "הארנב והצב"

הדמיה

6



החוליה בה מתחיל המעגל

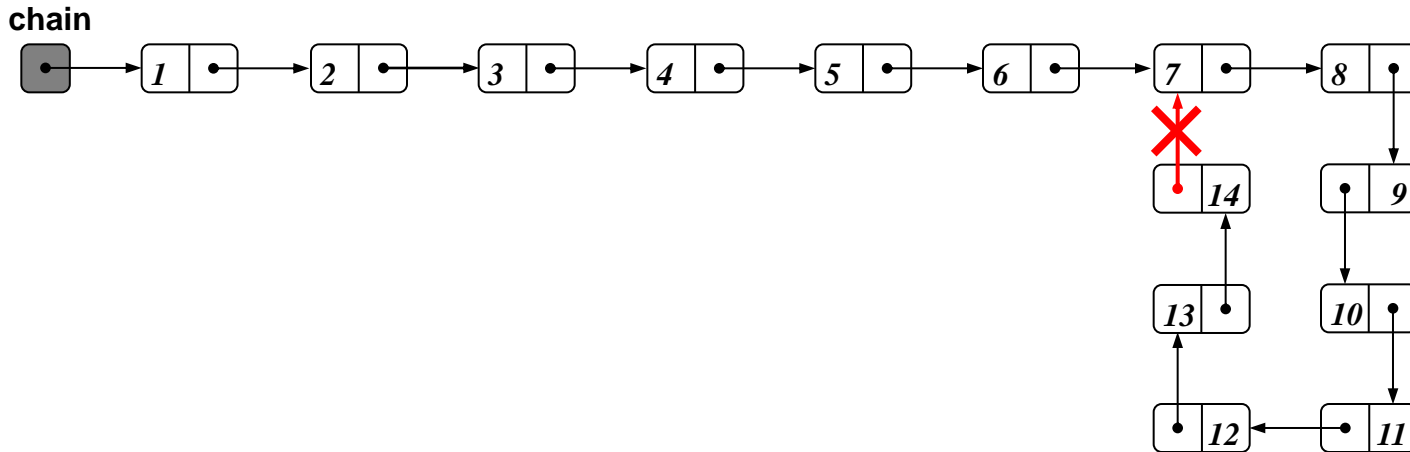
אלגוריתם "הארנב והצב"

שאלות נוספות:

1. כתבו פעולה המקבלת שרשרת החוליות מעגלית או ליניארית, ומחזירה אורך השרשרת.

2. כתבו פעולה המקבלת שרשרת החוליות מעגלית או ליניארית, ומדפיסה את הערכים השמורים בחוליות וביניהן המפריד " -> ". כמו-כן צריך לציין מעגל בשרשרת על-ידי סוגריים.

לדוגמה:



1->2->3->4->5->6->(7->8->9->10->11->12->13->14)

3. כתבו פעולה המקבלת שרשרת החוליות מעגלית או ליניארית, ומנתקת אותה מהמעגל ללא שום שינוי חוץ מניתוק ההפניה.

אלגוריתם "הארנב והצב"

שאלת אתגר:

4. קוד "הארנב והצב" שהוצג בכיתה מזהה האם יש מעגל בשרשרת חוליות, ומשתמש לשם כך בזיכרון בגודל קבוע, שאינו תלוי בשרשרת החוליות. תלמיד חסר סבלנות הציע להאיץ את ביצוע הקוד, באופן הבא:

בהינתן פרמטרים שהם מספרים טבעיים $0 < s < f$, הצב יבצע בכל איטרציה s התקדמויות לאורך שרשרת החוליות, והארנב יבצע f התקדמויות. הקוד המתאים להצעה מוצג כאן. שימו לב כי עבור $s=1$ ו- $f=2$ אנו מקבלים קוד שקול (אך כתוב באופן שונה במקצת) לקוד המקורי מההרצאה.

אלגוריתם "הארנב והצב"

שאלת אתגר:

```
public static <T> boolean detectCycle(Node<T> lst, int s, int f){
    Node<T> fast = lst;
    Node<T> slow = lst;
    do {
        for (int i = 0; i < f; i++)
            if (!fast.hasNext())
                return false;
            else
                fast = fast.getNext();
        for (int i = 0; i < s; i++)
            slow = slow.getNext();
    } while (fast != slow);
    return false;
}
```

אלגוריתם "הארנב והצב"

שאלת אתגר:

אלו מהטענות הבאות מתקיימות ביחס לקוד שהוצע ע"י אותו תלמיד חסר סבלנות נכונות או אינן נכונות? הצדיקו את תשובתכם ע"י מתן הסבר קצר או דוגמה נגדית.

- א. לכל זוג מספרים טבעיים $0 < s < f$ ולכל שרשרת חוליות lst (מעגלית או קווית), הקוד עוצר ומחזיר תשובה נכונה. לכל זוג מספרים טבעיים $0 < s < f$ ושרשרת חוליות lst הקוד החדש מבצע פחות השמות $next$ מהקוד המקורי.
- ב. יש זוגות מספרים טבעיים $0 < s < f$ ושרשרת חוליות קוויות lst , עבורן הקוד עוצר ומחזיר תשובה שגויה.
- ג. יש זוגות מספרים טבעיים $0 < s < f$ ושרשרת חוליות מעגליות lst , עבורן הקוד אינו עוצר.
- ד. לכל זוג מספרים טבעיים $0 < s < f$ ולכל שרשרת חוליות lst (מעגלית או קווית), הקוד עוצר ומחזיר תשובה נכונה. לכל זוג מספרים טבעיים $0 < s < f$ ושרשרת חוליות lst הקוד החדש מבצע יותר השמות $next$ מהקוד המקורי.