שאלה 1

1. Consider the directed acyclic graph G in Figure 3.10. How many topological orderings does it have?

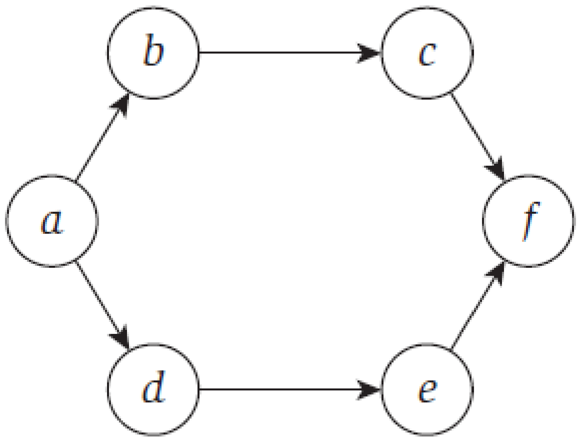
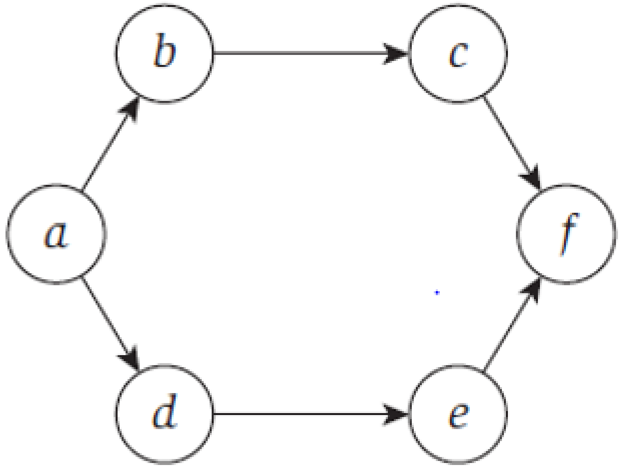


Figure 3.10 How many topological orderings does this graph have?

תשובה 1



צומת A חייב להיות ראשון, צומת F אחרון.
המקום היחיד לבחור הוא בצמתי הביניים.
שם b חייב לבוא לפני c ו-d לפני f.

לכן הסידורים האפשריים הם :

abcdef
abdcef
abdecf
adebcf
adbecf
adbcef

סך הכל שישה סידורים.

שאלה 2

2. Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, then your algorithm should output one. (It should not output all cycles in the graph, just one of them.) The running time of your algorithm should be $O(m + n)$ for a graph with n nodes and m edges.

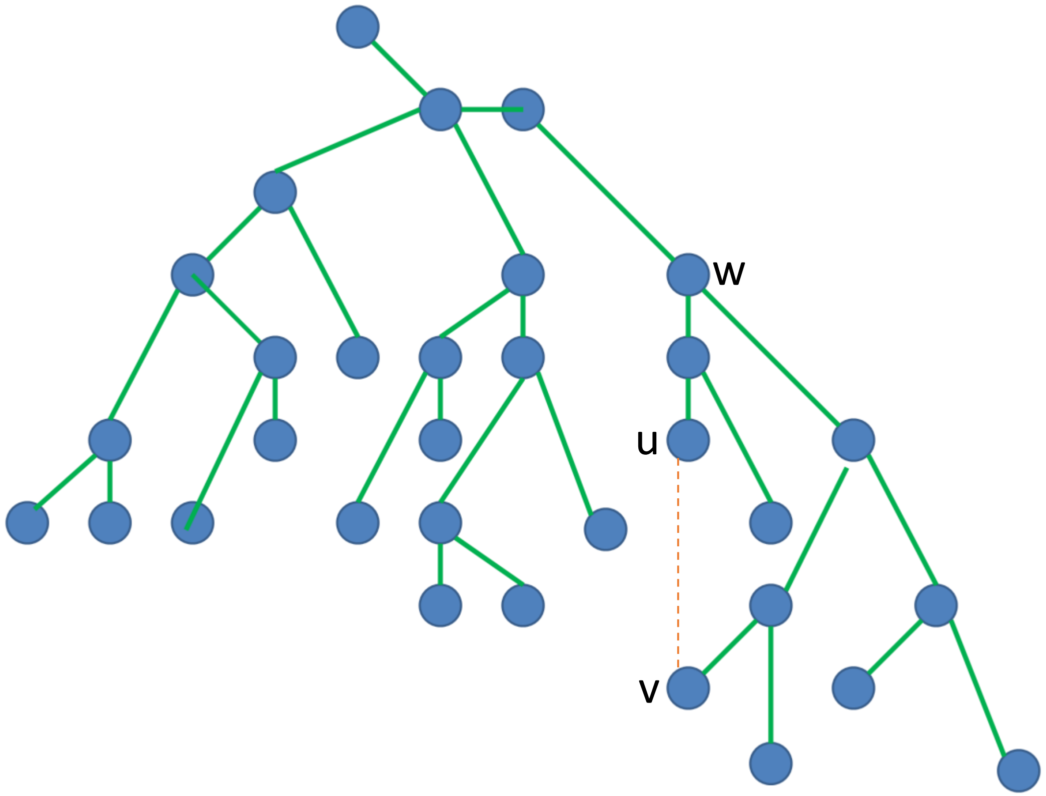
תשובה 2

נפרק את הגרף לרכיבי הקשירות שלו ונעבוד על כל רכיב קשירות בנפרד.

נריץ BFS על הגרף ונקבל עץ BFS.

אם כל קשתות הגרף נמצאות בעץ, הוא עץ בעצמו ולכן אין בו מעגלים.

אם יש קשת בגרף שאינה נמצאת בעץ ה-BFS, נניח u-v, נחפש בעץ ה-BFS את האב הקדמון הנמוך ביותר של שני קצות הקשת, נקרא לו w. כעת המסלולים v-w ו – u-w ביחד עם הקשת u-v מהווים מעגל.



שאלה 3

3. The algorithm described in Section 3.6 for computing a topological ordering of a DAG repeatedly finds a node with no incoming edges and deletes it. This will eventually produce a topological ordering, provided that the input graph really is a DAG.

But suppose that we're given an arbitrary graph that may or may not be a DAG. Extend the topological ordering algorithm so that, given an input directed graph G, it outputs one of two things: (a) a topological ordering, thus establishing that G is a DAG; or (b) a cycle in G, thus establishing that G is not a DAG. The running time of your algorithm should be $O(m + n)$ for a directed graph with n nodes and m edges.

תשובה 3

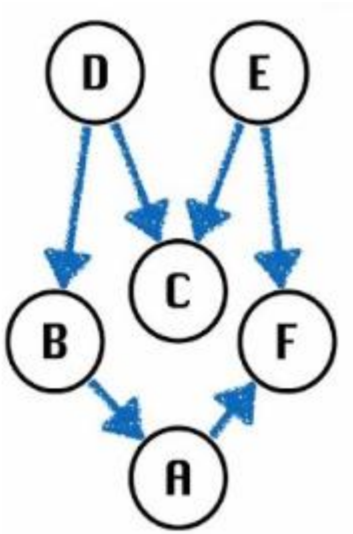
נפעיל את האלגוריתם למציאת מיון טופולוגי. אם הצלחנו – הרי שהגרף הוא DAG כלומר אין בו מעגלים.

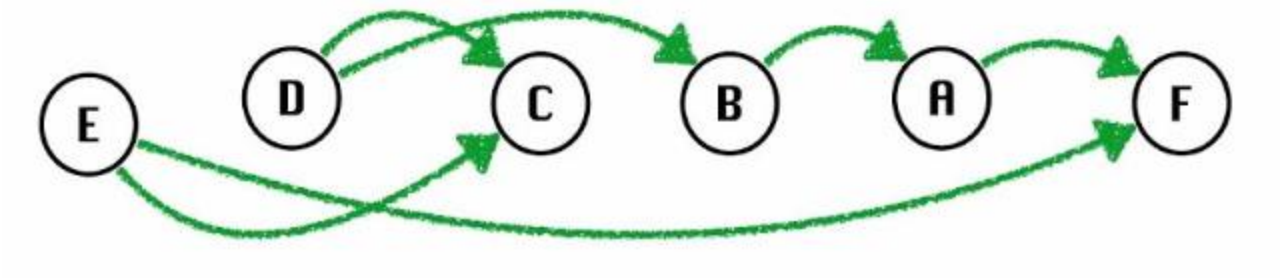
אם לא הצלחנו הרי שנתקענו בגלל שלכל אחד מהצמתים שנשארו יש דרגת כניסה לפחות 1.

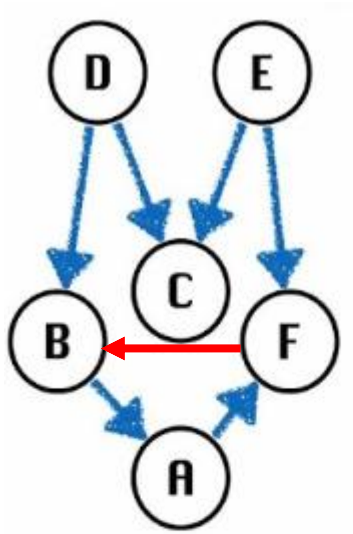
נבחר צומת ממנו נתחיל, ומשם והלאה נבצע שוב ושוב :
בחר קשת שנכנסת לצומת הנוכחי (בהכרח יש כזאת, כי דרגת הכניסה של כל הצמתים היא לפחות 1). לך נגד כיוון הקשת אל הצומת שבקצה השני שלה.

נמשיך לעשות כך עד שנחזור אל אחד הצמתים בהם כבר ביקרנו במהלך ביצוע הלולאה. בהכרח יש כזה כיון שהגרף הוא סופי.

אוסף הצמתים בהם עברנו בין שני הביקורים בצומת מהווה מעגל.







שאלה 4

4. Inspired by the example of that great Cornellian, Vladimir Nabokov, some of your friends have become amateur lepidopterists (they study butterflies). Often when they return from a trip with specimens of butterflies, it is very difficult for them to tell how many distinct species they've caught—thanks to the fact that many species look very similar to one another.

One day they return with n butterflies, and they believe that each belongs to one of two different species, which we'll call A and B for purposes of this discussion. They'd like to divide the n specimens into two groups—those that belong to A and those that belong to B—but it's very hard for them to directly label any one specimen. So they decide to adopt the following approach.

For each pair of specimens i and j, they study them carefully side by side. If they're confident enough in their judgment, then they label the pair (i, j) either "same" (meaning they believe them both to come from the same species) or "different" (meaning they believe them to come from different species). They also have the option of rendering no judgment on a given pair, in which case we'll call the pair *ambiguous*.

So now they have the collection of n specimens, as well as a collection of m judgments (either "same" or "different") for the pairs that were not declared to be ambiguous. They'd like to know if this data is consistent with the idea that each butterfly is from one of species A or B. So more concretely, we'll declare the m judgments to be *consistent* if it is possible to label each specimen either A or B in such a way that for each pair (i, j) labeled "same," it is the case that i and j have the same label; and for each pair (i, j) labeled "different," it is the case that i and j have different labels. They're in the middle of tediously working out whether their judgments are consistent, when one of them realizes that you probably have an algorithm that would answer this question right away.

Give an algorithm with running time $O(m + n)$ that determines whether the m judgments are consistent.

תשובה 4 - **רדוקציה של הבעייה לגרף**

נתרגם את הנתונים לגרף לא מכוון, ועליו נריץ אלגוריתם מוכר שייתן תשובה לשאלה.

עבור כל דגימה ניצור בגרף צומת. עבור כל שיפוט ניצור בגרף קשת שמחברת בין שני הצמתים הרלוונטיים. על כל קשת נציין מה כותרת השיפוט : האם הזוג דומים או שונים.

נפרק את הגרף לרכיבי הקשירות שלו ונעבוד על כל רכיב קשירות בנפרד.

נבחר צומת אקראי, נצמיד לו את הסימון (האקראי) A ונתחיל לבצע ממנו BFS. עם כל קשת שבעזרתה נגלה צומת חדש, נסמן את הצומת החדש בהתאם לכותרת שרשומה על הקשת : אם הכותרת היא "דומים" אזי על הצומת החדש נרשום את אותו סימון כמו בצומת ממנו הגענו אליו. אם הכותרת היא "שונים" אזי על הצומת החדש נרשום את הסימון השונה (A או B) מהצומת ממנו הגענו.

לאחר ה-BFS נעבור פעם נוספת על הגרף כולו, קשת אחרי קשת, ונבדוק : האם בכל קשת שכותרתה הינה "דומים" שני הצמתים שהיא מחברת הם בעלי אותו סימון ? האם בכל קשת שכותרתה הינה "שונים? הצמתים שהיא מחברת הם בעלי סימון שונה ?

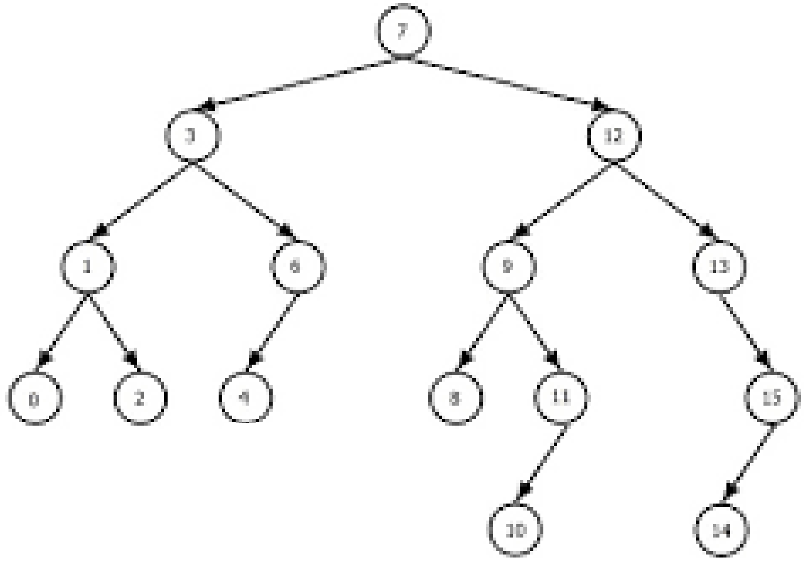
אם נמצא חוסר עקביות אפילו על קשת אחת, המשמעות היא שהשיפוטים אינם עקביים. אם כל הקשתות עקביות - כך הם גם השיפוטים.

```
For each component C of G
   designate a starting node s and label it A
   Mark s as ''Visited.''
   Initialize R=s.
   Define layer L(0)=s.
   For i=0,1,2,...
      For each node u in L(i)
         Consider each edge (u,v) incident to v
         If v is not marked ''Visited'' (then v is also not labeled)
            Mark v ''Visited''
            If the judgment (u,v) was ''same'' then
               label v the same as u
            else (the judgment was ''different'')
               label v the opposite of u
            Endif
            Add v to the set R and to layer L(i+1)
         Endif
      Endfor
   Endfor
Endfor
For each edge (u,v) (for each judgment (u,v))
   If the judgment was ''same''
      If u and v have different labels
         there is an inconsistency
      Endif
   Else (the judgment was ''different'')
      If u and v have the same labels
         there is an inconsistency
      Endif
   Endif
Endfor
```

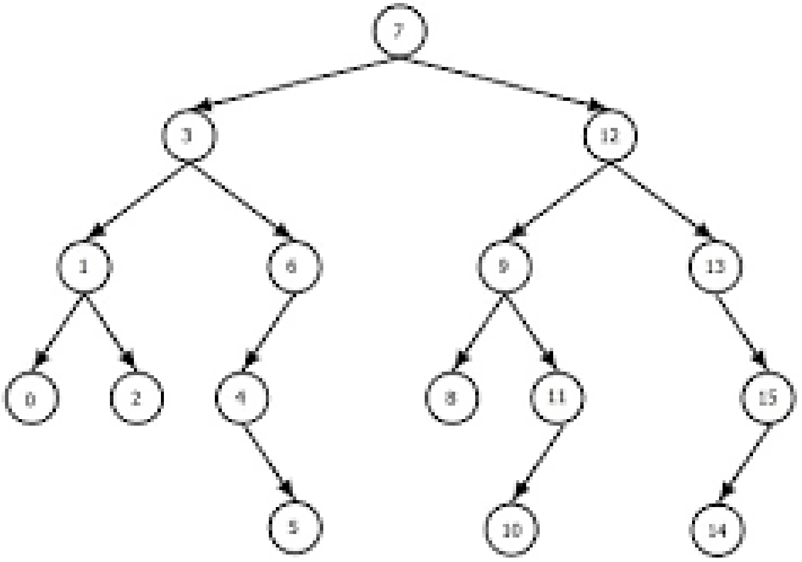
שאלה 5

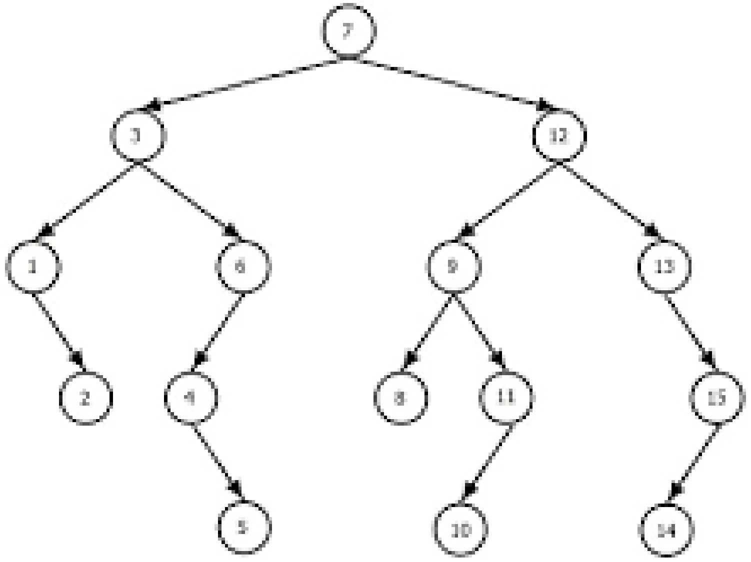
5. A binary tree is a rooted tree in which each node has at most two children. Show by induction that in any binary tree the number of nodes with two children is exactly one less than the number of leaves.

תשובה 5 - **הוכחה באינדוקציה על גודל הגרף**

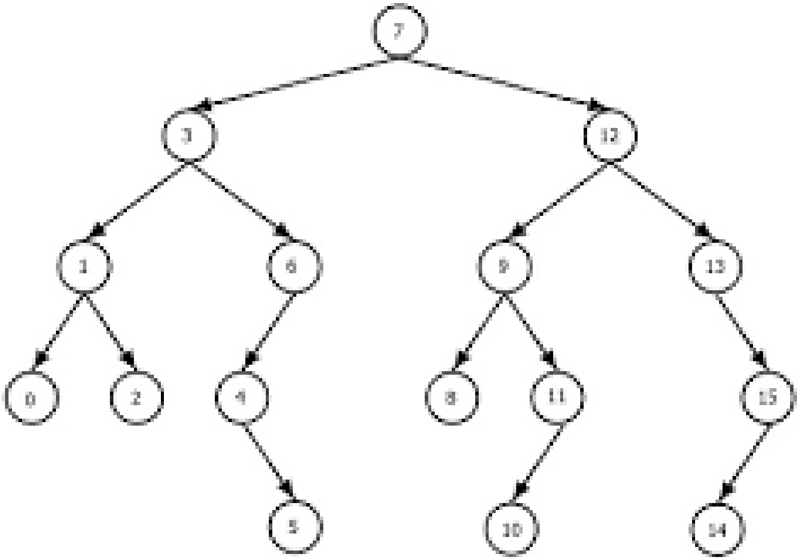


הוספת צומת כבן לצומת שהיה עלה





הוספת צומת כבן לצומת שכבר היה לו בן



שאלה 6

6. We have a connected graph $G = (V, E)$, and a specific vertex $u \in V$. Suppose we compute a depth-first search tree rooted at u, and obtain a tree T that includes all nodes of G. Suppose we then compute a breadth-first search tree rooted at u, and obtain the same tree T. Prove that $G = T$. (In other words, if T is both a depth-first search tree and a breadth-first search tree rooted at u, then G cannot contain any edges that do not belong to T.)

תשובה 6 - <u>הוכחה על דרך השלילה</u>

נניח שהגרף G אינו עץ, כלומר יש בו קשת e שאינה מופיע בעץ T.

נביט על העץ T כעל עץ DFS.

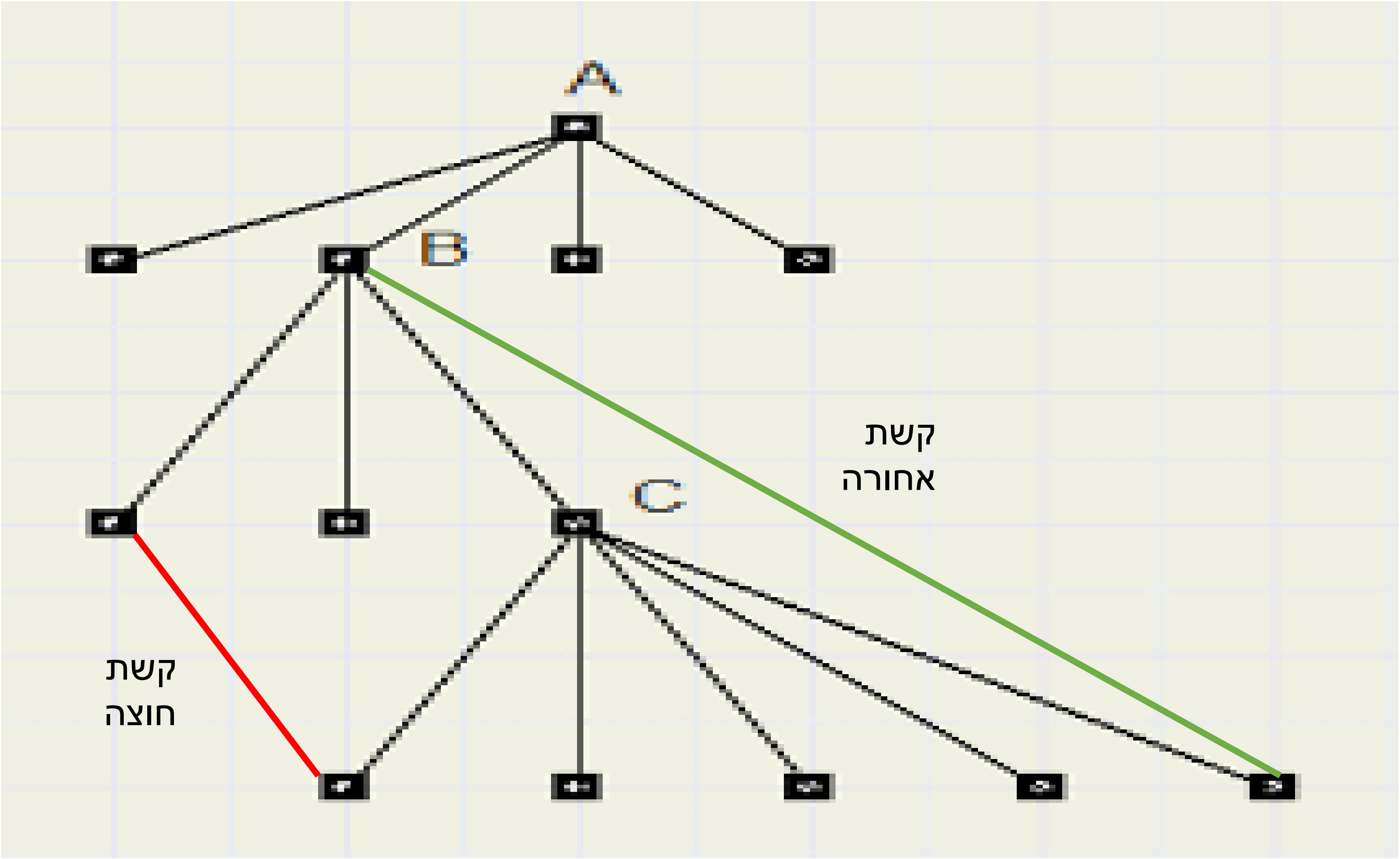
קשתות הגרף שאינו מופיעות בעץ ה-DFS הן בהכרח קשתות אחורה (תכונה של עץ DFS), ולא קשתות חוצות.

לכן הקשת e היא קשת אחורה שמחברת בין צומת לבין אב קדמון שלו (אך לא בין צומת לבין אבא מיידי שלו, כיון שלא תיתכנה קשתות מקבילות בגרף בלתי מכוון.)

נביט עתה על העץ T כעל עץ BFS.

כל קשתות הגרף בהכרח מחברות בעץ ה-BFS בין צמתים שנמצאים או באותה שיכבה או בשכבות סמוכות זו לזו (תכונה של עץ BFS).

אבל הקשת e (שהיא כזכור קשת מהגרף G) מחברת בין שני צמתים שנמצאים בשכבות שאינן סמוכות זו לזו.



סתירה ! לכן הגרף G הוא עץ, כלומר העץ T.

שאלה 7

7. Some friends of yours work on wireless networks, and they're currently studying the properties of a network of n mobile devices. As the devices move around (actually, as their human owners move around), they define a graph at any point in time as follows: there is a node representing each of the n devices, and there is an edge between device i and device j if the physical locations of i and j are no more than 500 meters apart. (If so, we say that i and j are "in range" of each other.)

They'd like it to be the case that the network of devices is connected at all times, and so they've constrained the motion of the devices to satisfy the following property: at all times, each device i is within 500 meters of at least $n/2$ of the other devices. (We'll assume n is an even number.) What they'd like to know is: Does this property by itself guarantee that the network will remain connected?

Here's a concrete way to formulate the question as a claim about graphs.

Claim: Let G be a graph on n nodes, where n is an even number. If every node of G has degree at least $n/2$, then G is connected.

Decide whether you think the claim is true or false, and give a proof of either the claim or its negation.

תשובה 7

בהינתן גרף G עם התכונה הנתונה, נניח על דרך השלילה שהוא אינו קשיר.

נביט ברכיב הקשירות הקטן ביותר שלו. בהכרח מספר הצמתים ברכיב הקשירות הזה הוא לכל היותר $n/2$.

נבחר צומת ברכיב הקשירות הזה. בהכרח לצומת זה יש דרגה שהיא לכל היותר מספר הצמתים האחרים ברכיב הקשירות שלו, כלומר $n/2 – 1$.

אך זה עומד בסתירה לנתון שדרגת כל צומת בגרף היא לפחות $n/2$.
מכאן שלא ייתכן שהגרף אינו קשיר.

שאלה 9

9. There's a natural intuition that two nodes that are far apart in a communication network—separated by many hops—have a more tenuous connection than two nodes that are close together. There are a number of algorithmic results that are based to some extent on different ways of making this notion precise. Here's one that involves the susceptibility of paths to the deletion of nodes.

Suppose that an n-node undirected graph $G = (V, E)$ contains two nodes s and t such that the distance between s and t is strictly greater than $n/2$. Show that there must exist some node v, not equal to either s or t, such that deleting v from G destroys all s-t paths. (In other words, the graph obtained from G by deleting v contains no path from s to t.) Give an algorithm with running time $O(m + n)$ to find such a node v.

תשובה 9

נריץ BFS החל מצומת s. נניח ש-t שוכן בשיכבה d של BFS זה.
לפי הנתון $d > n/2$.

נניח שבכל אחד מ-d השכבות של עץ ה-BFS יש לפחות שני צמתים.
אם כך בכל d השכבות יש לפחות $2*d$ צמתים, אך כאמור $d > n/2$, וקיבלנו שמספר הצמתים בכל d השכבות גדול מ-n, וזה לא ייתכן.

מכאן שבעץ ה-BFS יש לפחות שיכבה אחת שבה יש בדיוק צומת אחד, נקרא לו w.
ידוע שלא ייתכן שקשת בגרף המקורי תחבר בין שני צמתים שנמצאים בשתי שכבות שאינן סמוכות זו לזו (תכונה של BFS).

מכאן שכל מסלול בין s ל-t חייב לעבור בצומת w, ולכן מחיקתו מהגרף תנתק כל מסלול בין s ל-t.

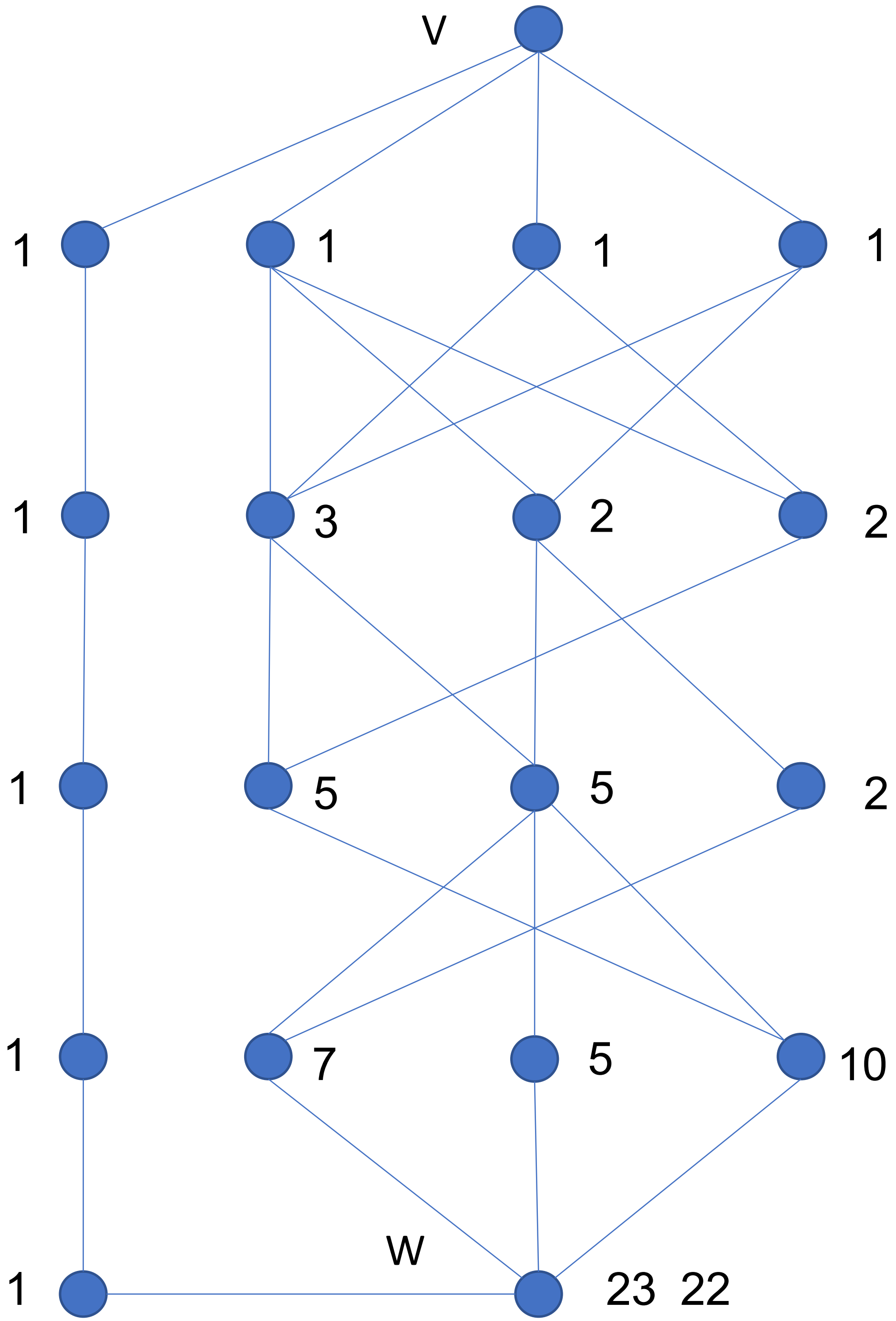
שאלה 10

10. A number of art museums around the country have been featuring work by an artist named Mark Lombardi (1951–2000), consisting of a set of intricately rendered graphs. Building on a great deal of research, these graphs encode the relationships among people involved in major political scandals over the past several decades: the nodes correspond to participants, and each edge indicates some type of relationship between a pair of participants. And so, if you peer closely enough at the drawings, you can trace out ominous-looking paths from a high-ranking U.S. government official, to a former business partner, to a bank in Switzerland, to a shadowy arms dealer.

Such pictures form striking examples of *social networks*, which, as we discussed in Section 3.1, have nodes representing people and organizations, and edges representing relationships of various kinds. And the short paths that abound in these networks have attracted considerable attention recently, as people ponder what they mean. In the case of Mark Lombardi's graphs, they hint at the short set of steps that can carry you from the reputable to the disreputable.

Of course, a single, spurious short path between nodes v and w in such a network may be more coincidental than anything else; a large number of short paths between v and w can be much more convincing. So in addition to the problem of computing a single shortest v-w path in a graph G, social networks researchers have looked at the problem of determining the *number* of shortest v-w paths.

This turns out to be a problem that can be solved efficiently. Suppose we are given an undirected graph $G = (V, E)$, and we identify two nodes v and w in G. Give an algorithm that computes the number of shortest v-w paths in G. (The algorithm should not list all the paths; just the number suffices.) The running time of your algorithm should be $O(m + n)$ for a graph with n nodes and m edges.



תשובה 10 - **שינוי של אלגוריתם מוכר**

נפתור את הבעייה הכללית של מציאת מספר המסלולים הקצרים ביותר בין v לבין כל הצמתים בגרף.

נריץ BFS החל מצומת v. אחת מתכונות ה-BFS היא שהוא מוצא את המסלולים הקצרים ביותר (בספירת קשתות) בין הצומת בו התחלנו לבין כל צומת אחר בגרף.

נגדיר S(x) להיות מספר המסלולים הקצרים ביותר בין v לבין הצומת x.
עבור כל צומת שנמצא בשיכבה הראשונה, זו שסמוכה ל-v, ברור ש- S(x) = 1.

עבור כל צומת אחר y שנמצא בשיכבה j מתקיים שהמסלול הקצר ביותר בין v לבינו הוא מהצורה הבאה : קודם כל מסלול בין v לבין צומת כלשהו x שנמצא בשיכבה j-1, ואז עוד קשת נוספת שמחברת בין x לבין y.

לכן S(y) הוא הסכום של כל S(x) עבור כל ה-x-ים שנמצאים בשיכבה אחת מעל זו שבה נמצא y.

את הסכומים הללו אפשר לחלחל במורד השכבות תוך כדי ביצוע ה-BFS, ולכן הסיבוכיות הינה בדיוק הסיבוכיות של BFS.