תרגיל בית #6:

שאלה 5. הוכח / הפרך : אם הגרף G עץ, הרצה של BFS והרצה של DFS יתנו את אותה תוצאה.
ברור שלא. תלוי בבחירות שהאלגוריתם עושה.

שאלה 6. הגדרה: יהי G = (V, E) גרף לא מכוון. רכיב קשיר ב- G הוא קבוצה של קודקודים $C \subseteq V$ מקסימלית, כך שלכל זוג קודקודים $u, v \in C$ קיים מסלול ב- G בין u ו- v.
הוכיחו כי ביער מסדר n (כלומר n קודקודים) עם k רכיבי קשירות יש n-k צלעות.
(הערה: בצורה אינטואיטיבית, רכיבי הקשירות של יער הוא אוסף העצים המכילים את היער).
באינדוקציה על k.
צעד – נסתכל על רכיב קשירות בגודל n1. יש לו n1-1 קשתות (כי הוא עץ). בשאר היער יש k-1 רכיבי קשירות וע"פ ה.א. n-n1-k+1 = (n-n1)-(k-1) קשתות ← סה"כ יש n-k = n-n1-k+1+n1-1.

שאלה 7. יהי G = (V, E) גרף מכוון ללא מעגלים. בהינתן 2 צמתים s,t, הצע אלגוריתם המחשב את מספר המסלולים הקצרים ביותר ביניהם.
שאלה לא מוגדרת נכון ולכן נפסלה מהבדיקה.
הכוונה הייתה למצוא את מספר המסלולים בין 2 צמתים ולא את מספר המסלולים הקצרים ביותר.
הפתרון:

```
Get-number-of-paths(s,t)
  If s=t return 1
  Count = 0
  For each u \in Adj[s]
        Count += Get-number-of-paths(u,t)
  Return count
```

לא נכנסים ללולאה אינסופית מכיוון שאין מעגלים.

תרגיל בית #7:

שאלה 2. בהינתן עץ AVL בו כל מפתח אינו מופיע יותר מפעם אחת נגדיר את הפעולה Cut(T,k):
אם k לא בעץ, הפעולה לא עושה דבר;
אם k בעץ, הפעולה מפרקת את העץ ל- 3 עצים באופן הבא:
(1) עץ חיפוש בינארי T1 המכיל את כל הקודקודים הקטנים מ- k.
(2) עץ חיפוש בינארי T2 המכיל את כל הקודקודים הגדולים מ- k.
(3) צמת המכיל את k.
א. תארו אלגוריתם המממש את Cut, כאשר העצים הנוצרים צריכים להיות עצי AVL . סיבוכיות הזמן הנדרשת היא O(n) במקרה הגרוע.
עוברים עם InOrder לתוך מערך. יוצאים עץ עם InOrder של כל הקודקודים שקטנים מ-k, ועץ עם הקודקודים שגדולים מ-k (על עצים כמעט שלמים).
ב. תארו אלגוריתם המממש את Cut, כאשר העצים הנוצרים לא בהכרח עצי AVL (אך כן עצי חיפוש בינאריים). סיבוכיות הזמן הנדרשת היא O(log n) במקרה הגרוע.
בשני הסעיפים הניחו כי n הוא מספר הקודקודים בעץ והסבירו את נכונות האלגוריתם והעמידה בדרישת הסיבוכיות.
מחפשים את k בעץ. אם הוא בעץ, קיבלנו 2 עצים בינאריים (שני ילדיו) כאשר אחד קטן מ-k ואחד גדול. נעלה למעלה לשורש. כל צמת v, נסמן את תת העץ שלה (שאינו הבו שהגענו דרכו) ב-Tv. מתקיים שאו ש-v,Tv קטנים מ-k או גדולים מ-k. במקרה שהם קטנים, נחבר את העץ הקטן מ-k שמצאנו בשלב הקודם ל-v (כאח של Tv). במקרה ששניהם גדולים, נחבר אותם לתת העץ הגדול מ-k.

שאלה 3. עבור הטענות הבאות, לכל טענה קבע אם היא נכונה או לא והוסף נימוק לנכונות הטענה / דוגמא נגדית.
א. כל עץ AVL הוא עץ חיפוש בינארי.
נכון.
ב. כל עץ חיפוש בינארי הוא מאוזן.

ברור שלא (שרוך ממויין)

ג. כל עץ מאוזן הוא עץ AVL.

לא נכון. AVL הוא סוג של עץ מאוזן.

ד. אחרי כל הכנסה של איבר חדש לעץ AVL חייבים לתקן את העץ.

כמובן שלא.

שאלה 4. נתונים שני עצי AVL T1 ו-T2 וקודקוד v כך ש-v קטן מכל איברי T1 וגדול מכל איברי T2.

א. הראו כיצד ניתן ליצור עץ AVL T המכיל את T1, T2, v באופן יעיל. ניתן להניח כי h(T1) ≥ h(T2).

פתרנו בכיתה. מחפשים קודקוד ברמה x - h(T2). מוסיפים לו בן v, ל-v מחברים את הבן של x ואת T2.

ב. כיצד תבצעו את האיחוד מבלי הקודקוד v.

אותו רעיון רק שמשתמשים בעלה המקסימאלי במקום ב-v.

שאלה 5. בהינתן שני עצי חיפוש בינאריים T1,T2 בעלי אותם n מפתחות, הראו כיצד ניתן לעבור מ-T1 ל-T2 תוך שימוש בלכל היותר 2n+O(1) רוטציות (פעולות Zig-Zig, Zi-g-Zag,Zig).

ניתן לעבור מכל עץ לשרוך ומכל שרוך לעץ ב-O(n) פעולות. לכן, עוברים מ-T1 לשרוך, ומהשרוך ל-T2.