Tree building pseudo – code

משפט. סדרת מספרים $d_1,d_2,\dots,d_{n-1},d_n$, d_i – מספר שלם חיובי ($d_i \in N$) מייצגת קדקודי העץ אם ורק אם מתקיים שוויון

$$\sum_1^n d_i = 2(n-1) \quad (*).$$

הכרחיות. אם T הוא עץ אז מספר קדקודיו גדול באחד ממספר צלעותיו וסכום דרגות הקדקודים שווה לפעמיים מספר צלעות.. לכן $\sum_1^n d_i = 2*|E| = 2(n-1)$.

מספקיות. נתונה סדרת מספרים $d_1,d_2,\dots,d_{n-1},d_n$, ($d_i \in N$) המקיימת את השוויון $\sum_1^n d_i = 2(n-1)$. צריך להוכיח כי הסדרה מייצגת קדקודי העץ.

הוכחה. באינדוקציה לפי מספר הקדקודים.

<u>בסיס</u>. n=1, אז $d_1=0$: ● - לעץ יש ר קדקוד אחד ודרגתו שווה 0 והתנאי מתקיים.

ניקח לדוגמה גם n=2 ●——● $d_2=d_1=1$, התנאי מתקיים

<u>הנחת אינדוקציה</u> הטענה נכונה עבור $n \geq 3$ כלשהו.

<u>שלב האינדוקציה</u> נוכיח את הטענה עבור n+1 ($\sum_1^{n+1} d_i = 2n$).

נמיין את מערך הדרגות בסדר יורד, מגדול לקטן: $d_1 \geq d_2 \geq \dots \geq d_{n-1} \geq d_n \geq d_{n+1}$

ידועה שלכל עץ יש לפחות שני עלים. אז בטוח: $d_{n+1}=1$, ו- $d_1 \geq \mathbf{2}$. בונים סדרה חדשה בגודל n :

$d_1-1,d_2,\dots,d_{n-1},d_n$, הסדרה החדשה מקיימת את התנאי:

$$d_1-1+d_2+\dots+d_{n-1}+d_n = d_1+d_2+\dots+d_{n-1}+d_{n+1}-2 = 2n-2 = 2(n-1)$$

ולפי הנחת האינדוקציה ניתן לבנות עץ שדרגתם הם $d_1-1,d_2,\dots,+d_{n-1},d_n$.

עכשיו נבנה עץ בעל n+1 קדקודים: נוסיף לקדקוד 1 עלה (קדקוד מספר n+1), דרגה שלו יגדל ב-1,

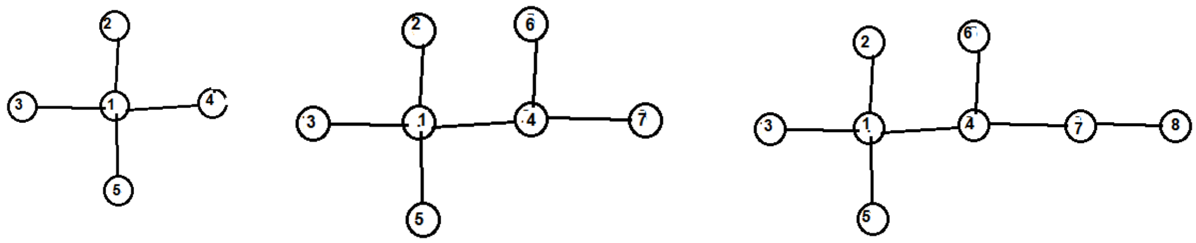
וסכום הדרגות יהי $\sum_1^{n+1} d_i = 2n$ = $d_1-1+1+d_2+\dots+d_{n-1}+d_n+1$. ■

יצירת העץ לפי הדרגות הנתונות ומקיימות את התנאי (*).

נמיין את מערך הדרגות בסדר יורד, מגדול לקטן: $d_1 \geq d_2 \geq \dots \geq d_{n-1} \geq d_n$.

בונים כוכב, שדרגת מרכז הכוכב הוא d_1, ניקח עלה אחד ונהפוך אותו למרכז הכוכב בדרגה d_2, ניקח עלה אחד ונהפוך אותו למרכז הכוכב בדרגה d_3, ונחזור לפעולה זו עד של לא נבנה את כל קדקודי החץ.

דוגמא: יהי n=8, 4,3,2,1,1,1,1,1 סדרת הדרגות, המקיימת את תנאי (*).



(a) (b) (c)

Input: Array of degrees: d [n]: $\sum_{1}^{n} d_i = 2(n-1)$, n – number of vertices

Output: tree – graph as adjacency matrix

```
public static ArrayList<Integer>[] treeBuilding(int deg[]){
      int n = deg.length;
      ArrayList<Integer>[] tree = new ArrayList[n];
      for (int i = 0; i < n; i++) {
            tree[i] = new ArrayList<Integer>();
      }
      int first_1 = 0;
      while(deg[first_1]>1) first_1++;
      int vertex = 0, numV = 1;
      if (n >= 3){
            while(vertex < first_1) {
                  for (int j = 0; j < deg[vertex]; j++) {
                        tree[vertex].add(numV);
                        tree[numV].add(vertex);
                        numV++;
                  }
                  vertex++;
                  deg[vertex]--;
            }
      }
      else if (n==2){
            tree[0].add(1);
            tree[1].add(0);
      }
      return tree;
}
```