5.1. Introduction
A sequence \((d_1,d_2,\ldots,d_n)\) of non-negative integers is called graphical if there’s at least one graph on vertices \(\{1,2,\ldots,n\}\) such that vertex \(K\) has degree \(d_k\) (Knuth, 2009).

The sequence \((1, 2,\ldots, n)\) are countable integers. A sorted sequence \(d_1\geq d_2\geq\ldots\geq d_n\) to become a graphic sequence for a simple graph, it has to satisfy the constraints such as \(d_1<n\) and the sum \(m=d_1+d_2+\ldots+d_n\) of any graphical sequence is always even and number of odd degree is even. Furthermore, it is easy to see that the sequence \(5,5,4,3,2,1\) satisfies the above constraints. However, no graph with above sequence of degrees exists. Therefore graphical sequences must also satisfy additional conditions. What are they?

There are two well-known necessary and sufficient conditions for a sequence of non-negative integers to be graphical: one was given independently by Havel (Havel, 1955 and Hakimi, 1962) while the other is a well-known explicit characterization due to Erdos and Gallai (Erdos and Gallai, 1960).

5.2. Preliminaries

**Theorem 1** (Havel-Hakimi) (Hakimi, 1962, Kim et al., 2008, Arumugam and Ramachandran, 2002):- There exists a simple graph with degree sequence \(d_1\geq d_2\geq\ldots\geq d_n\geq 0\) (\(n\geq 3\)) if and only if there exists one with degree sequence \(d_2-1,\ldots,d_{d_1+1}-1, d_{d_1+2},\ldots, d_n\).


Let \(d_1\geq d_2\geq\ldots\geq d_n>0\) be integers. Then they are the degree sequence of a simple graph if and only if

i) \(d_1+d_2+\ldots+d_n\) is even.

ii) for all \(K=1\) to \(n-1\) we have \(\sum_{i=1}^{K} d_i \leq K(K-1)+\sum_{i=K+1}^{n} \min(K, d_i)\)
Theorem 1 provides an algorithm to generate an actual graph with the given degree sequence while Theorem 2 only an existence result.

G. Sierksma and H. Hoogveen (Sierksma and Hoogeveen, 1991) identified the seven criteria for integer sequences being graphic. Theorem 1 gives an effective algorithm to determine whether a given partition \( P \) is graphical and to obtain a graph \( G \) realizing \( P \) when it is graphical.

**Algorithm:** Let \( P = (d_1, d_2, \ldots, d_n) \) be a partition of even integer \( p - 1 \geq d_1 \geq d_2 \geq \ldots \geq d_n \). \( P \) is graphical iff the following procedure results in a partition with every summand zero.

i) Identify the modified partition \( P' \) from Theorem 1.

ii) Sort the terms of \( P' \).

iii) Determine the modified partition \( P'' \) and sort.

iv) Continue this process as long as non-negative summands can be obtained.

If a partition obtained at an intermediate stage is graphical then \( P \) itself is graphical.

Assume that \( P' \) is graphical sequence. Then there exists a graph \( G_1 \) of order \( n-1 \). Such that \( P' \) is the degree sequence of \( G_1 \). Thus the vertices of \( G_1 \) can be labeled as \( v_2, v_3, \ldots, v_n \); so that

\[
\text{deg}(v_i) = \begin{cases} 
  d_{i-1}; & 2 \leq i \leq d_1 + 1 \\
  d_i; & d_1 + 2 \leq i \leq n 
\end{cases}
\]

A new graph \( G \) can be now constructed by adding a new vertex \( v_1 \) and the \( d_1 \) edges \( v_1v_i; 2 \leq i \leq d_1 + 1 \). Then in \( G \) \( \text{deg}(v_i) = d_i \) for \( 1 \leq i \leq n \), and so \( P = d_1, d_2, \ldots, d_n \) is graphical.

Conversely, let \( P \) be a graphical sequence. Hence, there exist graphs of order \( n \) with degree sequence \( P \). Among all such graphs let \( G \) be one, Such that \( V(G) = \{v_1, v_2, \ldots, v_n\}; \text{deg}(v_i) = d_i \) for \( i = 1, 2, \ldots, n \) and the \( \sum d_i = \) even number, the sum of degrees of the vertices adjacent with \( v_1 \) is maximum. We show first that \( v_1 \) is adjacent with vertices having degrees \( d_2, d_3, \ldots, d_{d_1+1} \).
Suppose to the contrary, that $v_1$ is not adjacent with vertices having degrees $d_2, d_3, \ldots, d_{d_1+1}$. Then there exist vertices $v_r$ and $v_s$ with $d_r > d_s$ such that $v_1$ is adjacent to $v_s$, but not to $v_r$. Since, the degree of $v_r$ exceeds that of $v_s$, there exists a vertex $v_t$ such that $v_t$ is adjacent to $v_r$ but not to $v_s$. Removing the degrees $v_1v_s$ and $v_rv_t$ and adding the edges $v_1v_r$ and $v_sv_t$, results in a graph $G'$ having the same degree sequence as $G$. However, in $G'$ the sum of the degrees of the vertices adjacent to $v_1$ is larger than that in $G$, contradicting the choice of $G$. Thus $v_1$ is adjacent with vertices having degrees $d_2, d_3, \ldots, d_{d_1+1}$ and the degree sequence $P'$. So $P$ is graphical.

We state a lemma after realizing the above theorems.

5.3. Algorithm Outline

**Lemma1**: For a graphic sequence $P=d_1, d_2, \ldots, d_n$ with $d_1 \geq d_2 \geq \ldots \geq d_n$; $n \geq 2$; $d_1 \geq 1$, if $d_n = 1$, then $(n-1) > d_1$ and $(n-1) > d_2$, where $n$ is number of vertices of the graph $G$.

Proof: - Let a simple graph $G (V, E)$ with $|V| = n$ number of vertices and $|E| = q$ number of edges. Also considering the degree sequence of the graph $G$ is $P+d_1, d_2, \ldots, d_n$ with $d_1 \geq d_2 \geq \ldots \geq d_n$; $n \geq 2$; and $d_n = 1$. That means the graph $G$ contains a pendent vertex. Now if we delete the pendent vertex from $G$, then the modified simple graph $G'$ will contain $(n-1)$ vertices. Then the degree of any vertex of $G'$ will not be $(n-1)$ and can be at most $(n-2)$. Now further adding the pendent vertex back to the graph $G'$ to produce $G$, cause degree of any one or more pendent vertex. Hence the lemma proved.

Graphic Integer Sequence (GlnS) with labeling is a universal representation of graphs. So, this is an universal superset of all other graph representation.

We have also identified the several criteria for integer sequences being graphic for a simple graph as shown below.

i) $d_1 \leq (n-1)$; for all $i=1,2,\ldots, n$.

ii) $\sum d_i$ = even number

iii) Number of odd degree vertices is even.
iv) For all $K=1$ to $n-1$ we have $\sum_{i=1}^{K} d_i \leq K(K-1)+ \sum_{i=K+1}^{n} \min(K, d_i)$

v) A partition $P=(d_1,d_2,\ldots,d_n)$ of even number into $n$ parts with $n-1 \geq d_1 \geq d_2 \geq \ldots \geq d_n$ is graphical if the modified partition $P'=(d_2-1,d_3-1,\ldots,d_{d_1+1}-1,d_{d_1+2},\ldots,d_n)$ is graphical.

vi) For an integer sequence $P=d_1,d_2,\ldots,d_n$ with $d_1 \geq d_2 \geq \ldots \geq d_n$; $n \geq 2$; $d_1 \geq 1$, if $d_n=1$, then $(n-1) > d_1$ and $(n-1) > d_2$, where $n$ is number of vertices of the graph $G$.

vii) $\sum_{i=1}^{K} d_i \leq (k)(n-m-1) + \sum_{i=n-m+1}^{n} d_i$ for each $k \leq n$, $m \geq 0$ and $k + m \leq n$.

On the above, we have imagined the given degree sequence as a collection of Stubs: at each vertex $i$ there is $d_i$ edges; anchored at the vertex, but the other ends are free. Connecting two stubs at two distinct nodes will form an edge between those nodes. During our procedure, we will call the residual degree the number of current stub of node.

The Havel-Hakimi algorithm for constructing a graph realizing a graphical sequence $P$ works as follows: Connect all stubs of the node with the largest residual degree to nodes that have the largest residual degree and repeat until no stubs are left. It is easy to see that the Havel-Hakimi algorithm cannot create all simple graphs realizing the sequence; instead it creates a graph in which high degree nodes tend to be connecting other high degree nodes.

We have tried to study different characteristic of a graph using the Graphic integer sequence as a decision maker.
5.4. Proposed algorithms

We have proposed different algorithms that accept a graphic integer sequence as the deciding factor.

i) Random graph generation.

ii) Random regular graph generation.

iii) Determine whether given integer sequence represents at least one tree or forest.

iv) Existence of Hamiltonian graph or an Euler graph.

v) Connectedness.

vi) Decision of Maximum Clique size.

vii) Automorphic graph generation.

viii) Anonymous graph generation etc.

We have also identified different graph theoretic problems those can be solved using Graphic integer sequence. So, Graphic integer sequence becomes the canopy of the graph.

The general form of algorithms for tracking different graph theoretic problems using graphic integer sequence.

Algorithm: General algorithms for tracking different graph theory problems.

Step1. Identify whether the given integer sequence is graphic.

Step2. Do apply different problem specific constraints on this sequence.

Step3. Generate the random graph satisfying the specific constraint.

Step4. Stop.
5.5. When an integer sequence represents a tree sequence (P-5)

5.5.1. Introduction

**Definition:** A sequence \( d_1, d_2, d_3, \ldots, d_n \) of non-negative integers is said to be a *tree sequence* if \( \xi \) represents a *graphic sequence* and there exists a tree \( T \) whose vertices have degree \( d_i \) and \( T \) is called realization of \( \xi \).

For an example:

\( \xi = 2,2,2,1,1 \) is a sequence of nonnegative integers. This \( \xi \) represents a graphic sequence (Arumugam and Ramachandran, 2002, Chartrand and Lesniak, 1996). This sequence also represents a tree sequence.

![Figure 3](image-url)

**Theorem 1:** If the average degree of a connected graph \( G \) be greater than two, \( G \) has at least two cycles (Hartsfield and Ringle, 1997).

**Proof:** (Hartsfield and Ringle, 1997).

Let us assume \( G \) be connected graph and also assume that \( d_1, d_2, d_3, \ldots, d_n \) be the degree sequence of \( G \). Then if the average degree is greater than two, we have,

\[
2 < (d_1 + d_2 + d_3 + \ldots + d_n) / n
\]

Since,

\[
\sum d_i = 2e; \quad \forall i = 1,2,3,\ldots,n
\]

\[
\therefore 2e/n > 2.
\]

\[
\therefore e > n
\]

Let, in the limiting condition \( e = n + 1 \).
Since, connected graph of \(n\) vertices with \((n-1)\) edges, is tree (Chartrand and Lesniak 1996, Hartsfield and Ringle 1997, Deo 2006). Adding an extra edge to a tree forms a circuit (Chartrand and Lesniak 1996, Hartsfield and Ringle 1997, Deo 2006). Hence extra two edges in a tree form two circuits. Hence, the theorem proved.

**Lemma 1:** Let, a sequence \(d_1, d_2, d_3, \ldots, d_n; \) where \(d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n\) is the given degree sequence of graph \(G\), then the sequence \(\bar{\xi}\) is said to be a tree sequence, if and only if,

\[
\frac{\sum d_i}{n} = \text{average degree of vertices} < 2 \quad \text{................. (1)}
\]

and

\[
\frac{\sum d_i}{2} = e = (n-1) \quad \text{................................. (2)}
\]

**Proof:**
According to the Theorem 1, \(\frac{\sum d_i}{n} = \text{average degree of vertices} < 2\), if the graph \(G\) is connected then it is circuit free and also if \(e = (n-1)\) then the connected graph \(G\) is a tree. Conditions (1) and (2) are the necessary condition for a sequence to be a tree sequence, but not the sufficient conditions.

Let us consider a sequence \(\bar{\xi} = 2,2,2,1,1\) is a degree sequence of a graph. Now,
\[
\frac{\sum d_i}{n} = \frac{2+2+2+1+1}{5} = 1.6 < 2; \quad n = 5; \quad \{\text{condition (1) satisfied}\}
\]

and
\[
e = \frac{\sum d_i}{2} = \frac{2+2+2+1+1}{2} = 4;
\]
\[
\therefore e = (n-1); \quad \{\text{condition (2) is satisfied}\}. \text{Still, we are not sure. Because, the sequence } \bar{\xi} \text{ represents both a tree and disconnected graph.}
\]

---

**Figure 4:** (a) Disconnected graph                          (b) Tree
5.5.2. Foundation of the proposed Algorithm

We know that a tree has at least two pendent vertices (Arumugam and Ramachandran 2002, Deo, 2006). Now, if we delete a pendent vertex from a tree, it again represents another tree and hence it must have at least two pendent vertices (Arumugam and Ramachandran, 2002, Deo, 2006). Repeating this way we obtain a situation where only an edge remains.

Now deletion of a pendent vertex means deletion of a leaf vertex from a non-leaf vertex and this decreases the degree of the non-leaf vertex by unity. It can be shown that, deletion of a leaf vertex from a non-leaf vertex arbitrarily, is similar to deletion of a pendent vertex from a tree (vide Lemma 2). In our algorithm instead of arbitrary deletion we have deleted a pendent vertex and decreased the degree of the highest vertex by unity.

**Lemma 2:** Deletion of a pendent vertex from an arbitrary vertex is similar to deletion of a pendent vertex from highest degree vertex.

**Proof:**
Let we are given a Tree Sequence $\xi$. For this sequence there are two or more isomorphic trees possible. We also note that, all internal vertices of same number of child vertices have more degree than the root vertex. So we call always draw a tree with internal vertex having a pendent vertex. That will make the vertex of highest degree vertex of the tree.

We can delete a pendent vertex from any of the internal vertex, hence from the highest degree internal vertex. That will decrease the vertex degree by unity. Deletion of the pendent vertex from the tree gives another tree of different degree sequence $\xi'$. Same processes discussed above are true for $\xi'$. Hence the lemma proved.

The proposed theorem which is the basis of the Algorithm is now given below.
Theorem 2: Let a sequence $d_1, d_2, d_3, \ldots, d_n$; where $d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n$; $n \geq 2$. The sequence $\xi$ is said to be tree sequence if and only if $d_n = d_{n-1} = 1$ and $\xi = d_1 - 1, d_2, d_3, \ldots, d_{n-1}$ is a tree sequence.

Proof:
Let, $\xi$ be a tree sequence. Then there exists a tree $T_1$ with $(n-1)$ vertices, such that $\xi$ is the degree sequence of the tree $T_1$. Thus, the vertices of $T_1$ can be labeled as $V_1, V_2, \ldots, V_{n-1}$; so that

$$
\deg(V_i) = \begin{cases} 
  d_1 - 1; & i = 1 \\
  d_i; & 2 \leq i \leq n-1 
\end{cases}
$$

Adding a new vertex $V_n$ and a single edge can construct a new tree $T$ from $T_1$. That increases the $d_1$ degree by unity. Then in $T$ $\deg(V_i) = d_i$ for $1 \leq i \leq n$, and so $\xi = d_1, d_2, d_3, \ldots, d_n$ is the degree sequence of $T$. Hence, $\xi$ is a tree sequence. Hence, the theorem proved.

5.5.3. Proposed algorithm

In the next section, we are going to propose an algorithm for checking a given graphic sequence to be a tree sequence.

The proposed algorithm is TREE_SEQ.

Algorithm TREE_SEQ:

Input: Sequence of nonnegative integers.

Output: Valid or Invalid Tree sequence.

Step 1:
$\xi = d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n$. $i = 0$;

Step 2:
If $(d_n(0) = d_{n-1}(0) = 1)$ Then
    Continue;
Else
    Jump to Step 6.
End If
Step 3:
Delete $d_n$ from sequence and decrease $d_1$ by unity.
$$\xi^{(i+1)} = d_1^{(i)} - 1, d_2^{(i)}, d_3^{(i)}, \ldots, d_{n-1}^{(i)}$$
Sort the sequence to make
$$\xi^{(i+1)} = d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_{n-1}$$

**STEP 4:**
If ($\xi^{(i+1)} \neq \emptyset$ and $d_{n-i}^{(i)} = 1$) Then
$$i = i + 1$$
Repeat through Step 3.
Else
If ($\xi^{(i+1)}$ contains even number of $1$'s) Then
Print “The sequence $\xi$ as Tree sequence”
Stop.
Else
Jump to Step 6.
End If
End If

**Step 5:**
If ($\xi^{(i+1)} = \emptyset$) Then
Print “The sequence $\xi$ is Tree sequence”
Stop.
End If

**Step 6:**
If (The sequence $\xi^{(i+1)} \neq \emptyset$ and $d_{n-i}^{(i)} \neq 1$) Then
Print “The sequence $\xi$ is not a Tree sequence”
End If

**Step 7:**
Stop
5.5.4. Explanation of the Algorithm TREE_SEQ with an Example:

Let us consider a degree sequence $\xi = 4, 3, 3, 2, 1, 1, 1, 1, 1$ and have to check that the sequence represents a tree or not.

Now,

$\xi^{(0)} = 4, 3, 3, 2, 1, 1, 1, 1, 1$

Since,

$d_{10} = d_0 = 1$, we go for the next step.

Now, delete $d^{(0)}_{10}$, decrease max$(\xi^{(0)})$ i.e. $d_1$ by unity and sort $\xi^{(0)}$ to form $\xi^{(1)}$, same processes are repeated till the conditions are TRUE. We have,

$\xi^{(0)} = 4, 3, 3, 2, 1, 1, 1, 1, 1$

$\xi^{(1)} = 3, 3, 3, 2, 1, 1, 1, 1$

$\xi^{(2)} = 2, 3, 3, 2, 1, 1, 1, 1$

$\xi^{(3)} = 2, 3, 2, 2, 1, 1, 1$

$\xi^{(4)} = 2, 2, 2, 2, 1, 1$

$\xi^{(5)} = 1, 2, 2, 2$

$\xi^{(6)} = 1, 2, 2$

$\xi^{(7)} = 1, 2$

$\xi^{(8)} = 1, 1$

Since, $\xi^{(8)}$ contains even number of 1’s, we stop and conclude that “The Sequence is a tree sequence”.
5.5.5. Complexity and Experimental Results

In the proposed algorithm the space and time complexity is $O(n^2)$, where $n$ is the number of positive integers.

The experiment is carried out on real data. Each sequence is generated randomly and then checked whether it is a valid Tree Sequence or not. The program was run in a Pentium IV (1.33GHz) PC, Windows XP platform and in Borland C compiler.

Figure 5: The tree for the corresponding sequence $\xi = 4,3,3,2,1,1,1,1,1,1$. 
In the Table 1, the first column (No. of Vertices) indicates number of vertices for which random sequences are generated and the column Time indicates time required for checking the sequence whether it is Tree Sequence or not.
5.5.6. Conclusion:
The algorithm we proposed (in the Section 4.1) actually identifies that the given graphic sequence is tree sequence or not. Any two isomorphic graphs represent the exactly same sequence. However, the converse is not true (Arumugam and Ramachandran, 2002). So, the proposed theorem (Theorem 2, in Section 3) actually says that if the given graphic sequence is tree sequence then at least one tree can be possible for the sequence. Two non-isomorphic graphs, one is tree another is a disconnected graph; represent the same degree sequence (shown in the Figure 4).
5.6. Hamiltonian Circuit and Euler Trail (P-6)

5.6.1. Introduction

**Definition:** For any graph $G$, we define $\delta(G) = \min\{\deg(v) \mid v \in V(G)\}$ and $\Delta(G) = \max\{\deg(v) \mid v \in V(G)\}$.

If all the vertices of $G$ have the same degree $d$ then $\delta(G) = \Delta(G) = d$ and in this case the graph $G$ is called the *Regular graph* of degree $d$ (Arumugam and Ramachandran, 2002).

**Definition:** A *Hamiltonian circuit* (or *Hamiltonian Cycle*) in a graph $G$ is a circuit which contains every vertex of $G$ (Clark and Holton, 1995). A graph $G$ is called *Hamiltonian Graph* if it contains a Hamiltonian cycle (Clark and Holton, 1995).

**Definition:** A *trail* in $G$ is called an *Euler Trail* if it includes every edge of $G$ (Clark and Holton, 1995).

**Definition:** A *Spanning Sub graph* $G'$ of $G$ is a sub graph containing all the vertices of $G$ (Harary, 1988, Hartsfield and Ringle, 1997, Deo 2006).

**Theorem 1** (Dirac, 1952): If $G$ is a graph with $n \geq 3$ vertices and $\delta \geq n/2$, then $G$ is Hamiltonian (Arumugam and Ramachandran, 2002, Clark and Holton, 1995).

**Theorem 2:** A degree sequence of graph $G$ is given by $\xi = d_1, d_2, d_3, \ldots, d_n$. There is an at least one Hamiltonian graph if and only if the degree sequence of the graph $G$ can be represented as $\xi' = 2, 2, 2, \ldots, n^{th}$ term, where $n \geq 3$.

**Proof:** Hamiltonian circuit is a closed path, where each vertex papers once. Hence in a Hamiltonian Circuit degree of each vertex is exactly two.

Let us consider a given degree sequence $\xi$ of the graph $G$. Now, deletion of a edge from a graph causes decrease of degrees of two vertices by unity. Assuming, $\xi = d_1, d_2, d_3, \ldots, d_n$ and $d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n > 1$. If we can construct $\xi' = 2, 2, 2, \ldots, n^{th}$ term from $\xi$, that means $\xi'$ represents the degree sequence of the spanning sub graph of $G$. Hence there must be a sub graph of $G$ which is a circuit containing all the vertices of $G$, which is nothing but the Hamiltonian circuit of the graph $G$. Hence, the theorem is proved.
Theorem 3: A connected graph $G$ is Euler iff the degree of every vertex is even (Clark & Holton, 1995).

Theorem 4: A connected graph $G$ has an Euler Trail iff it has at most two odd degree vertices, i.e., it has either no vertex of odd degree or exactly two vertices of odd degrees (Clark and Holton, 1995).

5.6.2. Proposed Algorithm: HAMILTON_CKT

Input: Sequence array.

Output: Contains Hamiltonian Circuit or not..

Step 1:

$\xi(0) = d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n > 1$

IF ($d_n = 1$) Then
Stop.
End If

Step 2:

$k = d_1 - 1; d_1 = 2$
$d_2 = d_2 - 1, d_3 = d_3 - 1, \ldots, d_{k-1} = d_{k-1} - 1;$

Step 3:

Sort the sequence to make
$\xi(i) = d_1(i) \geq d_2(i) \geq d_3(i) \geq \ldots \geq d_n(i)$

Step 4:

If ($d_1 = d_2 = d_3 = \ldots = d_n = 2$) then Jump to Step 5
Else
If ($d_n = 1$) Then
Jump to Step 6
Else
Jump to Step 2
End If
End If

Step 5:

Print “The Graph is Hamiltonian Graph”
Stop.
Step 6:  
Print “The Graph is not a Hamiltonian Graph”

Step 7:  
Stop.

5.6.3. Explanation of the Algorithm HAMILTON_CKT with an Example

Let a degree sequence is $\xi = 3, 3, 2, 2, 2, 2, 2$. We have to check that the degree sequence represents a Hamiltonian graph or not.

Since, $n = 7$, and $d_7 \neq 1$ we go for the next Step.

$k = 3 - 2 = 1$; $d_1 = 2$, $d_2 = 3 - 1 = 2$

Now, $\xi' = 2, 2, 2, 2, 2, 2$. Here, $d_1 = d_2 = d_3 = \ldots = d_n = 2$ satisfied. Hence, we stop here.

![Figure 6: (a) (b) Not a Hamiltonian Graph. (c) A Hamiltonian Graph.](image-url)
5.6.4. Proposed Algorithm: EULER_TOUR

**Input:** Sequence array.

**Output:** Represents Euler graph or not.

**Step 1:**
\[ \xi(0) = d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n \]

**Step 2:**
If (\( \forall d_i \) is even numbers OR \( \xi \) contains exactly two odd numbers) Then
Print “The sequence represents a Euler graph”
Else
Print “The sequence represents a Euler graph”
End If

**Step 3:**
Stop.

5.6.5. Complexity

The complexity of the proposed algorithm of 5.6.2 is \( O(n^3) \) and the complexity of the proposed algorithm of 5.6.4 is \( O(n^2) \) where \( n \) is the number of positive integers.

5.6.6. Conclusion

The algorithms we proposed actually identifies that the given graphic sequence represents any Hamiltonian graph or an Euler graph or not. Any two isomorphic graphs represent the exactly same sequence. However, the converse is not true (Arunugam and Ramachandran, 2002). So, the proposed theorems HAMILTON_CKT and EULER_TOUR actually says that if the given graphic sequence is a sequence of a Hamiltonian Graph or Euler Graph respectively then at least one Hamiltonian Graph and an Euler Graph can be possible for the sequence. Three non-isomorphic graphs, among them one is Hamiltonian Graph, but they represent the same degree sequence (shown in the Figure 1).
5.7. Connectedness

5.7.1. Introduction

If the sequence $\xi = d_1, d_2, d_3, \ldots, d_n$, be a graphic sequence then is there any condition for which we can say that $\xi$ represents a connected or disconnected Graph.

Theorem 1: If $e > (n-1)C_2$ the simple graph is a connected graph (Bondy and Hemminger, 2006).

Proof: $e = (n-1)C_2$ represents total number of edges possible for a connected, simple (complete graph) $G$ with $(n-1)$ vertices. Now if $e > (n-1)C_2$ the extra edges must connect to the extra vertex with $G$ to form another simple graph $G'$ with $n$ vertices. Hence the graph is connected. Hence the theorem is proved.

Theorem 2: A graph $G$ with $n$ vertices and $\delta \geq (n-1)/2$ is connected (Arumugam and Ramachandran, 2002).

Proof: Suppose $G$ is not connected. Then $G$ has more than one component. Consider any component $G_1 = (V_1, E_1)$ of $G$. Let, $v_1 \in V_1$. Since, $\delta \geq (n-1)/2$ there exists at least $(n-1)/2$ vertices in $G_1$ adjacent to $v_1$ and hence $V_1$ contains at least $(n-1)/2 + 1 = (n+1)/2$ vertices. Thus each component of $G$ contains at least $(n+1)/2$ and $G$ has at least two components. Hence, number of vertices in $G \geq n+1$ which is a contradiction. Hence the theorem is proved.

5.7.2. Proposed Criteria

Now, we are producing seven necessary criteria for a given simple graph to be a connected or disconnected graph.

Criteria 1:

A graph $G$ with $n$ vertices and $e$ edges if $e < (n-1)$, then the graph is a disconnected graph.
Criteria 2:
Let $\xi = d_1, d_2, d_3, ..., d_n$ be the degree sequence of $G$ with $n$ number of vertices and $e$ number of edges. If $(n-1) \leq e \leq \frac{(n-1)C_2}{2}$ then at least one graph is connected for the sequence $\xi$.

Criteria 3:
Let $\xi = d_1, d_2, d_3, ..., d_n$ be the degree sequence of $G$ with $n$ number of vertices, $e$ number of edges, $\partial_r = \frac{(n-1)}{2}$ (required $\partial(G)$ for connectedness), $d_1 \neq (n-1)$ and $\partial_r \leq d_n$ implies that the sequence $\xi$ represents connected graph sequence.

Criteria 4:
Let $\xi = d_1, d_2, d_3, ..., d_n$ be the degree sequence of $G$ with $n$ number of vertices, $e$ number of edges, $\partial_r = \frac{(n-1)}{2}$ (required $\partial(G)$ for connectedness),
\[ \partial_r > d_n \text{ and } d_n = d_{n-1} = 1 \]
or
\[ \partial_r > d_n \text{ and } d_n \neq 1 \]
represents at least one graph can be possible with given sequence $\xi$ which is disconnected graph.

Criteria 5:
Let $\xi = d_1, d_2, d_3, ..., d_n$ be the degree sequence of $G$ with $n$ number of vertices, $e$ number of edges, $\partial_r = \frac{(n-1)}{2}$ (required $\partial(G)$ for connectedness), $\partial_r > d_n$ and $d_n = 1 < d_{n-1}$ implies that the sequence $\xi$ must be a connected graph sequence and no disconnected graph can be represented by the sequence.

Criteria 6:
Let $\xi = d_1, d_2, d_3, ..., d_n$ be the degree sequence of $G$ with $n$ number of vertices, $e$ number of edges. If $d_1 = (n-1)$ then $\xi$ represents a connected graph sequence.

Criteria 7:
Let $\xi = d_1, d_2, d_3, ..., d_n$ be the degree sequence of $G$ with $n$ number of vertices, $e$ number of edges. Number of 1’s in $\xi$ (which must be even number) is greater than $d_1$ and $\xi$ of $\xi$ then the sequence $\xi$ is connected sequence.
5.7.3. **Proposed theorem:**

Let $\xi$ be the degree sequence of given graph $G$. If $\xi = d_1 \geq d_2 \geq d_3 \geq \ldots, d_n$ then $\xi$ is said to be:

(i) **Disconnected graph sequence**

If $A' = A \odot C$ is graphic.
And
$B' = B - C$ is graphic
And
$B' \neq \emptyset$

(ii) **Connected graph sequence**

If $A' = A \odot C$ is graphic.
And
$B' = B - C = \emptyset$
Or
$B' = B - C \neq \emptyset$
And
$B'$ is not graphic.

Where, the symbol $\odot$ means the concatenation symbol

- $\xi = d_1 \geq d_2 \geq d_3 \geq \ldots, d_n$
- $\xi' = d_2 - 1 \geq d_3 - 1 \geq \ldots, d_n$
- $A = d_2 - 1 \geq d_3 - 1 \geq \ldots, d_{d_1 - 1}$
- $B = d_{d_1 + 2} \geq \ldots, d_n$
- $C = \{X_i \in B \mid I = d_1 + 2, \ldots, n}\}$
- $A' = A \odot C$ and $B' = B - C$.

**Proof:** Let, the sequence is $\xi = d_1 \geq d_2 \geq \ldots, d_n$ a collection of non-negative integers. Deleting $d_1$ and reducing $d_2, d_3, \ldots, d_{d_1 + 1}$ by unity to produce $\xi' = d_2 - 1 \geq d_3 - 1 \geq \ldots, d_n$. This is also collection of non-negative integers. Now, $\xi' = A \odot B$ where, $A = d_2 - 1 \geq d_3 - 1 \geq \ldots,$ $d_{d_1 - 1}$ and $B = d_{d_1 + 2} \geq \ldots, d_n$ also $C \subseteq B = \{X_i \in B \mid I = d_1 + 2, \ldots, n}\}$. 
Now,

**Case I**

A' and B' are two components of G. Because, successive reduction (Hakimi, 1962) makes A' toØ and B' toØ individually, without effecting each other.

So, the sequence ξ represents at least one graphic sequence which is disconnected graph.

**Case II**

B' implies ξ representing G, contains a single component. Hence the sequence ξ represents a connected graph sequence and no disconnected graph is possible for the sequence ξ.

### 5.7.4. Analysis with examples

**Example 1:**

Let ξ = (3,3,2,2,2,2,2)

ξ' = (2,2,2,2,1,1,1)

A = (2,2,2), B = (2,1,1) and C = (2,1,1)

A' = (2,2,2,2,1,1)

B' = Ø

Now, A' is graphic.

Finally C = Ø, A' = (2,2,2), B' = (2,1,1)

A' and B' both are graphic.

Then we say that ξ represents disconnected sequence.

**Example 2:**

Let G₀ = The theta graph of degree sequence of (3,3,2,2,2,2). Then ξ' = (2,2,2,1,1)

A = (2,1,1) and B = (2,2). A is graphic but B is not graphic. Then ξ cannot represent any disconnected sequence. This is true for the theta graph.
5.7.5. Complexity

The complexity of the proposed algorithm of 5.7.3. is $O(n^2)$ where $n$ is the number of positive integers.

5.7.6. Conclusion

The algorithms we proposed actually identifies that the given graphic sequence represents any connected or disconnected Graph sequence or not. Any two isomorphic graphs represent the exactly same sequence. However, the converse is not true (Arumugam and Ramachandran, 2002). So, the proposed theorem and criterions actually determines that if the given graphic sequence is a connected sequence or disconnected sequence or at least one Graph can be possible for the sequence (which is connected or disconnected).
5.8. Maximum Clique Number (P-7)

5.8.1. Introduction

**Definition 1:** A sequence $\xi$ has property $A_p$ iff there exists a graph with degree sequence $\xi$ in which the first $p$ vertices form a complete subgraph (Rao, 1976).

**Definition 2:** If $p < n$ and $(p-1) \leq d_1 \leq (n-1)$ then $p$-laying off $d_1$ from $\xi$ and reducing $d_2, d_3, d_4, \ldots, d_p$ and $(d_1-p+1)$ largest terms among $d_{p+1}, d_{p+2}, \ldots, d_n$ by unity (Rao, 1976).

**Definition 3:** $\xi$ has property $A_p$ iff the sequence obtained by $p$-laying off $d_1$ from $\xi$ has property $A_{p-1}$ (Rao, 1976).

**Definition 4:** If $\xi$ is the degree sequence of a graph $G$ and $G$ contains a maximum clique number $k$, then

$$m = \sum (d_i - k + 1)$$

implies reducing largest terms among $d_{k+1}, d_{k+2}, \ldots, d_n$ by unity.

**Theorem 1:** Let, a sequence $\xi = (d_1, d_2, d_3, \ldots, d_n)$ and let, a sequence $\xi^{p,k} = (d_1^{p,k}, d_2^{p,k}, \ldots, d_{n-k+1}^{p,k})$ be the sequence obtained from $\xi$ by $(p-k+1)$-laying off $d_1^{p,k}$ for $k = 2, 3, 4, \ldots, p$, where $\xi^{p,1} = \xi$. Then $\xi$ has the property $A_p$ if and only if $\xi^{p,1}$ is graphic (Rao, 1976).

**Proof:** (Rao, 1976)

This theorem gives a procedure to determine whether $\xi$ has property $A_p$ since it is easy to find out whether $\xi^{p,1}$ is graphic or not. This can be used to construct a
graph on $p$ vertices (when it exists) starting from a graph with degree sequence can also be determined easily.

**5.8.2. Illustration of the procedure with an example**

Consider a sequence, 
\[ \xi^{4,1} = 6, 5, 4, 4, 4, 4, 4, 1 \]

With $p = 4$, $d_1 = 6$ 
and since $p - 1 = 3 \leq d_1 \leq (n - 1)$ 
\[ \Rightarrow 3 \leq 6 \leq 7 \]

\[ \therefore \xi^{4,2} = 4, 3, 3, 3, 3, 3, 1 \]

Now, $d_1 = 4$ since, $p - 1 = 3 \leq d_1 \leq (n - 1)$ 
\[ \Rightarrow 3 \leq 4 \leq 7 \]

\[ \therefore \xi^{4,3} = 2, 2, 3, 2, 2, 1 \]

Now, $\xi^{4,4} = 1, 2, 2, 2, 1$

Now, $\xi^{4,4}$ is graphic. Working backward, it is easy to construct a graph with degree sequence in which 4 vertices form a complete sub graph.
Figure 7: The corresponding graph $\xi = 6, 5, 4, 4, 4, 4, 4, 1$.

With $p = 5$, $d_i = 6$
And since $p-1 = 4 \leq d_i \leq (n-1)$
$\Rightarrow 4 \leq 6 \leq 7$
$\therefore \xi^{5,2} = 4, 3, 3, 3, 3, 3, 1$

Now, $d_i = 4$ since, $p-1 = 4 \leq d_i \leq (n-1)$
$\Rightarrow 4 \leq 4 \leq 7$
$\therefore \xi^{5,3} = 2, 2, 2, 2, 3, 1$

Now, $\xi^{5,4} = 1, 1, 3, 2, 1$

Now, $\xi^{5,5} = 0, 3, 2, 1$

Since, $\xi^{5,5}$ is not graphic we conclude that maximum clique number cannot be 5 it must be 4 for the given degree sequence $\xi$.

5.8.3. Foundation of the proposed Algorithm

The problem is to find out the maximum clique number of a graph $G$, with a degree sequence $\xi$. Since maximum clique number is nothing but the maximum number of vertices forming complete sub graph of $G$. If $k$ be the maximum clique number of $G$ then in $\xi$ there must be at least $k$ number of integers in $\xi$ is $\geq k-1$.

So, we have taken $1^{st} k$ integers with value greater than equal to $k-1$ from $\xi$. And this represents a complete sub graph of $k$ vertices. Deleting this sub graph from $G$ we get $G'$ with total edges $E' = E\{k(k-1)/2 + m\}$ where $E$ is total number of edges of $G$, $m = \Sigma (d_i-k+1)$. Therefore we get $\xi'$ from $\xi$ after drop-$m$. If $\xi'$ represents a graphic sequence then $k$ is maximum clique number.
The proposed theorem which is the basis of the Algorithm is now given below.

**Theorem 2:** Let $\xi = d_1, d_2, d_3, \ldots, d_n$ be the degree sequence of a graph $G$. The graph $G$ contains a maximum clique number $k$ iff $d_1 \geq k + 1$, $k \leq n$ and the sequence after droop-m $(d_{k+1}, d_{k+2}, \ldots, d_n)$ is graphic.

**Proof:**
Let us consider $\xi$ represents a graphic degree sequence of graph $G$ and after drop-m we have $\xi'$ from $\xi$, which is not graphic sequence. Since, drop-m implies deletion of a complete subgraph of $k$ vertices from $G$ and reducing next $(d_{k+1}, d_{k+2}, \ldots, d_n)$ maximum $m$ terms by unity. Then after drop-m we have $\xi'$ from $\xi$. Since $\xi$ is graphic, after drop-m $\xi'$ must represent a graphic sequence. That contradicts our assumption. Hence the theorem is proved.

### 5.8.4. Proposed Algorithm:

The proposed algorithm MAX_CLQ is given in this section.

**Algorithm MAX_CLQ:**

**Input:** Sequence of nonnegative integers.

**Output:** $k$ maximum clique number.

**Step 1:**

\[ \xi = d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n, \ k = d_1; \]

**Step 2:**

If $(d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_k \geq (k-1))$ Then

Continue;

Else

Jump to Step 4.

End If

**Step 3:**

If $((d_{k+1}, d_{k+2}, \ldots, d_n)$ is graphic after drop-m) Then

Jump to Step 5.

Else

Continue.

End If

**Step 4:**

\[ k = k - 1 \]
**Step 5:**
Print “The maximum clique number of the sequence is $k$”

**Step 6:**
Stop.

### 5.8.5. Explanation of the Algorithm MAX_CLQ with an Example:

Let us consider a degree sequence $\xi = 6, 6, 4, 4, 4, 2, 2$ and have to find out the maximum clique number for the sequence, representing a graph $G$.

Now, Since $d_1 = 6$ therefore $k = d_1 = 6$.

Now, $d_2 \geq 5$, but $d_3 < 5$.

$\therefore k = k-1 = 5$

Now, $d_2 \geq d_3 \geq \ldots \geq d_5 \geq 4$

$\therefore m = \sum (d_i - k + 1) \ \forall i = 1, 2, \ldots 5$.

$\therefore m = 4$

$\therefore \text{drop-}m \text{ from } (d_6, d_7, d_8) \text{ we have }$

$\xi' = (2, 1, 1)$

Which is graphic.

$\therefore \text{The sequence } \xi \text{ contains a maximum clique number } 5.$
Figure 8: The corresponding graph $\xi=6, 6, 4, 4, 4, 4, 2, 2$

5.8.6. Complexity and experimental Results:
The complexity of the proposed algorithm in 5.8.4 is $O(n^3)$ where $n$ is the number of positive integers as well as number of vertices.
The experiment is carried out on real data. Each sequence is generated randomly and then find the maximum clique number and compared the results with existing one. The program was run in a Pentium IV(1.33GHz) PC, Windows XP platform and in Borland C compiler.

Table 2: Result

<table>
<thead>
<tr>
<th>No. of Vertices</th>
<th>Present</th>
<th>Previous[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0</td>
<td>6.758</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>6.758</td>
</tr>
<tr>
<td>300</td>
<td>0.11</td>
<td>6.7581</td>
</tr>
<tr>
<td>400</td>
<td>0.17</td>
<td>6.7581</td>
</tr>
<tr>
<td>500</td>
<td>0.33</td>
<td>7.1043</td>
</tr>
<tr>
<td>600</td>
<td>0.6</td>
<td>7.1046</td>
</tr>
<tr>
<td>700</td>
<td>0.94</td>
<td>7.4519</td>
</tr>
<tr>
<td>800</td>
<td>1.59</td>
<td>8.1445</td>
</tr>
<tr>
<td>900</td>
<td>2.2</td>
<td>8.8382</td>
</tr>
<tr>
<td>1000</td>
<td>3.46</td>
<td>9.8784</td>
</tr>
<tr>
<td>1100</td>
<td>4.01</td>
<td>11.2646</td>
</tr>
<tr>
<td>1200</td>
<td>5.54</td>
<td>13.3435</td>
</tr>
<tr>
<td>1300</td>
<td>6.76</td>
<td>15.7687</td>
</tr>
<tr>
<td>1400</td>
<td>8.62</td>
<td>18.5406</td>
</tr>
<tr>
<td>1500</td>
<td>10.71</td>
<td>22.3527</td>
</tr>
<tr>
<td>1600</td>
<td>12.31</td>
<td>26.8563</td>
</tr>
<tr>
<td>1700</td>
<td>15.27</td>
<td>32.0532</td>
</tr>
<tr>
<td>1800</td>
<td>17.85</td>
<td>55.2185</td>
</tr>
<tr>
<td>1900</td>
<td>18.4015</td>
<td>57.0873</td>
</tr>
<tr>
<td>2000</td>
<td>19.3884</td>
<td>61.9856</td>
</tr>
</tbody>
</table>

In the Table 2, the first column (No. of Vertices) indicates number of vertices for which random sequences are generated and the column Time indicates time
required for finding the maximum clique number and compared the results with existing one.

5.8.7. Conclusion:

The algorithm we proposed actually identifies that the given graphic sequence represents a maximum clique size graph possible. Any two graphs represent the exactly same sequence. However, the converse is not true (Arumugam and Ramachandran, 2002). So, the network topologies design purpose where reliability is major concern, there the proposed algorithm is more appropriate.
5.9. Applications of designated algorithms:

1. **Computer Networking:** A practical way to generate network topology that meet the observed data is the following degree driven approach: First predict the degrees of the graph by extrapolation from the available data, and then construct a graph meeting the degree sequence and possibly additional constraints such as connectivity or “randomness”, to name a couple. This is the most successful approach for modeling the topology of the internet.

2. **Chemistry:** Chemists may try to organize the whole structure of an organic molecule from knowledge of its decomposition products.

3. **Security:** Graph anonymous problem is related with social security where same identities are given to maximum number of vertex treating an individual person. Graphic integer sequence shows the one way to providing the security.

4. **Cryptography:** In graph, ambiguity is very much present until labeling are not concern. So, our proposed algorithms are very much in cohesion with the data hiding. Where graphs are used as key for data encryption or decryption purpose.

5. **Social Network:** Where connectivity, clustering and clique are the major areas to focus. The designated algorithms are relevant to the following purpose.
5.10. Conclusion:-

GInS is a novel idea that puts all existing intractable problems under a head. It shows that how the sequence can dilute the complexity of some landmark graph algorithms in a different outlook. The labeling information merged with the GInS represents the graph uniquely. Their combination gives the better advantages than the other representation of graph.