**<u>Colouring a graph containing no odd circles</u>**

Let $G = (N, E)$ be such a graph.

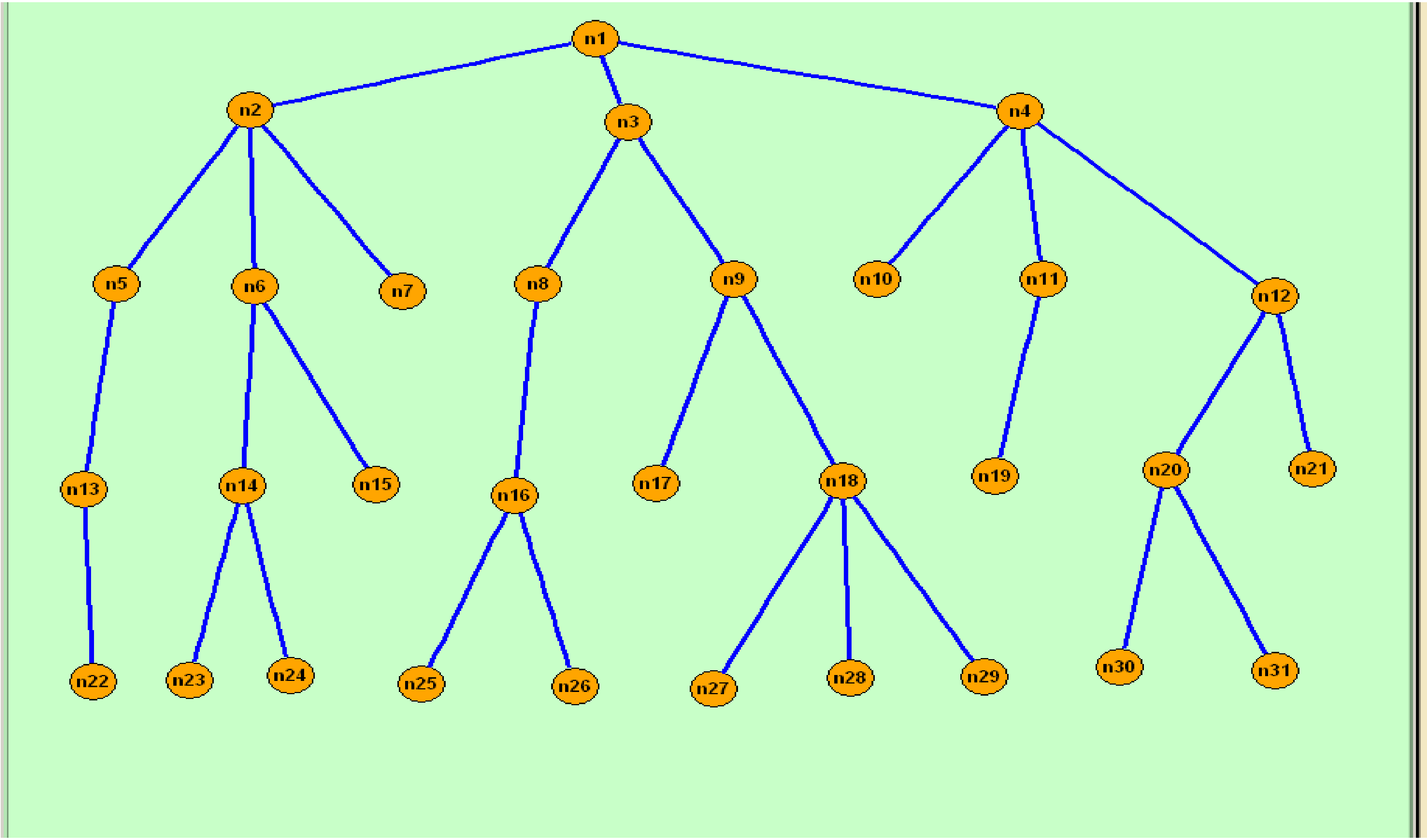
Start from any node $v_1$. Assign $v_1$ level 0.

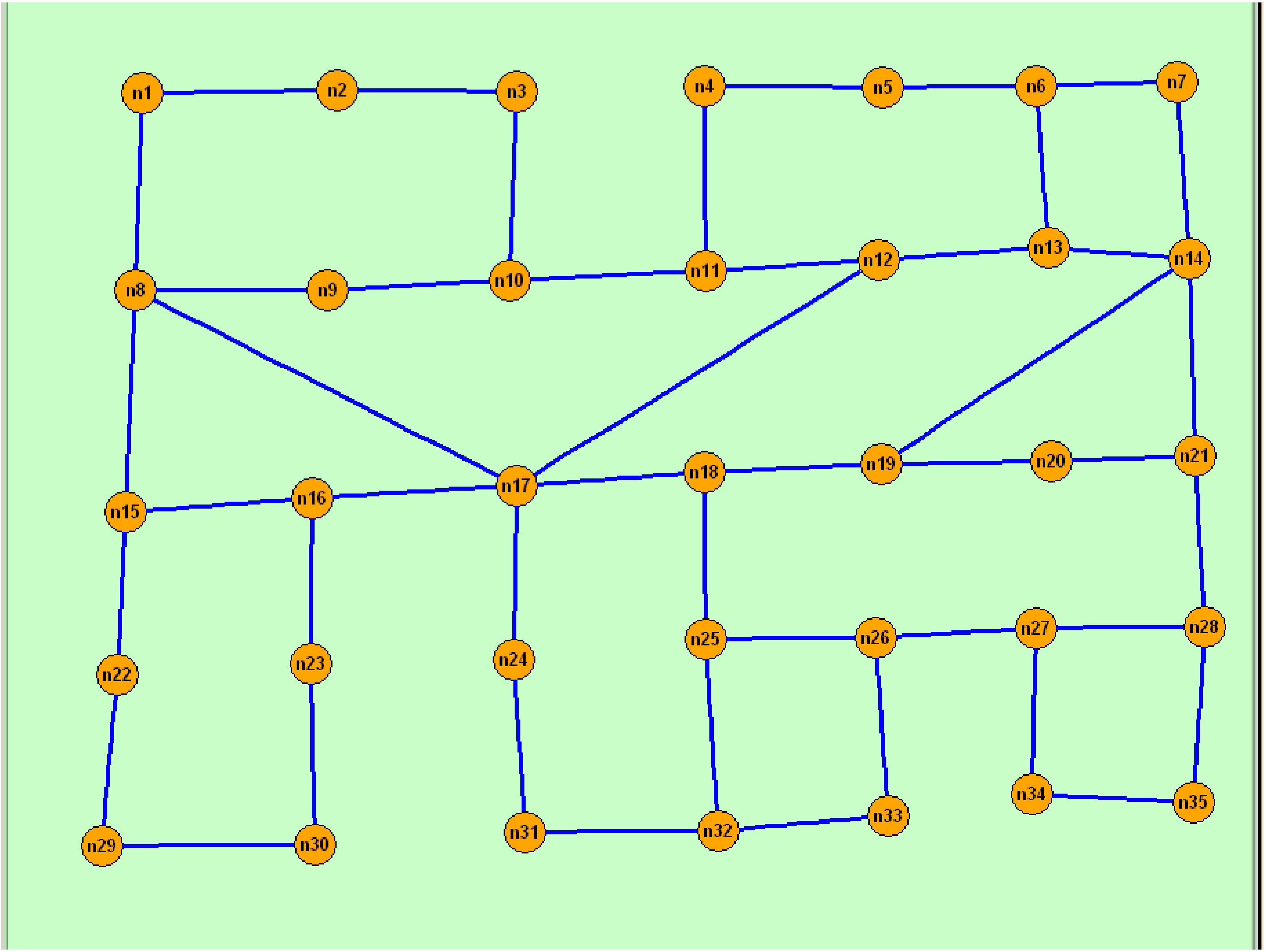
Assign to all nodes adjacent to $v_1$ level 1.

Partition the set of nodes of $G$ into sets of nodes at different levels.

Let $v_{i_1}, v_{i_2}, \ldots, v_{i_r},$ be the nodes at level $i$. Consider all nodes adjacent to $v_{i_1}$ that are not assigned levels 0, 1, 2, ..., $i$. Assign these level $i + 1$.

Continue this process until all nodes are assigned a level. Assign now one colour to nodes at odd levels and another colour to all those at an even level.

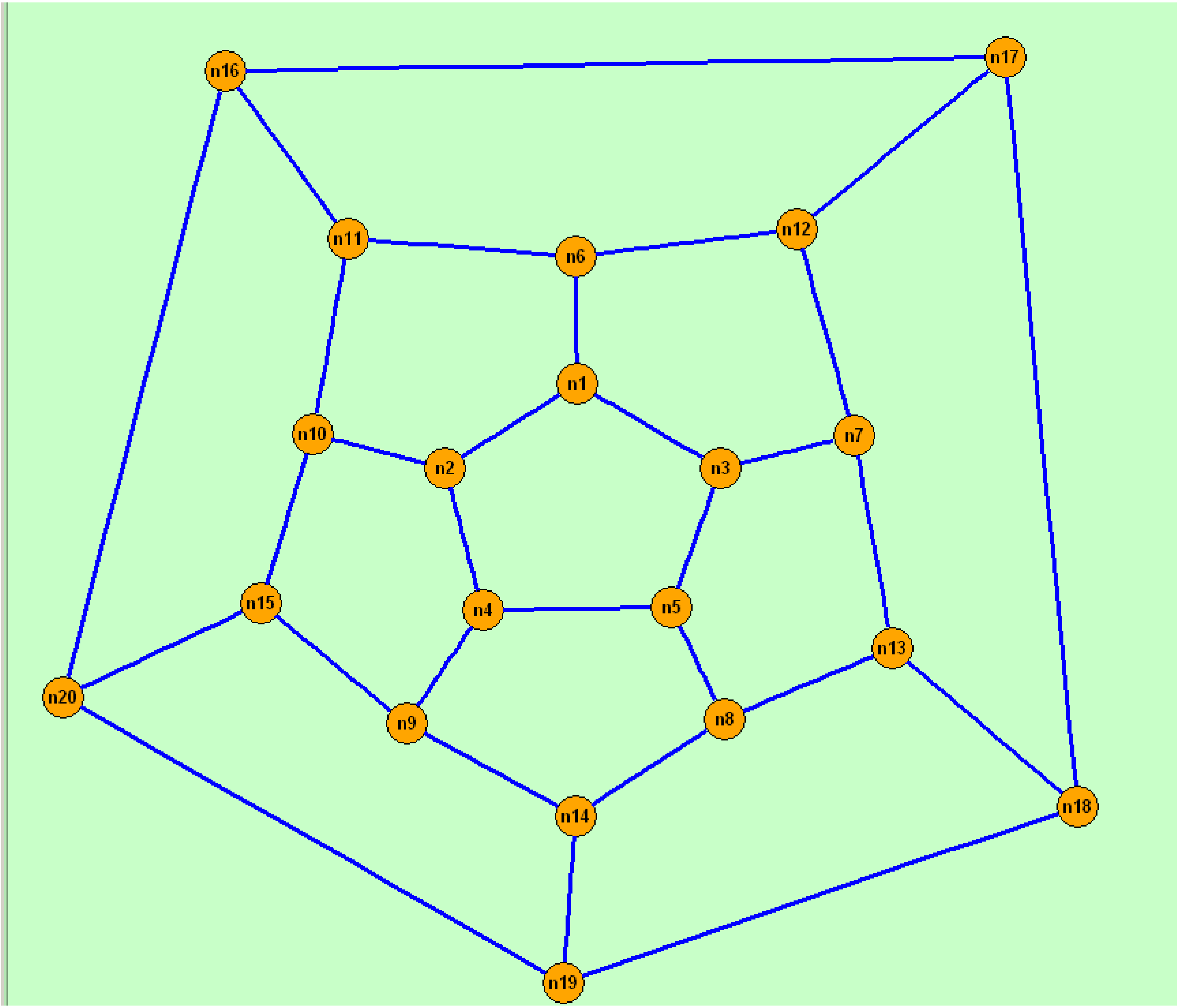

n1
n2
n3
n4
n5
n6
n7
n8
n9
n10
n11
n12
n13
n14
n15
n16
n17
n18
n19
n20
n21
n22
n23
n24
n25
n26
n27
n28
n29
n30
n31

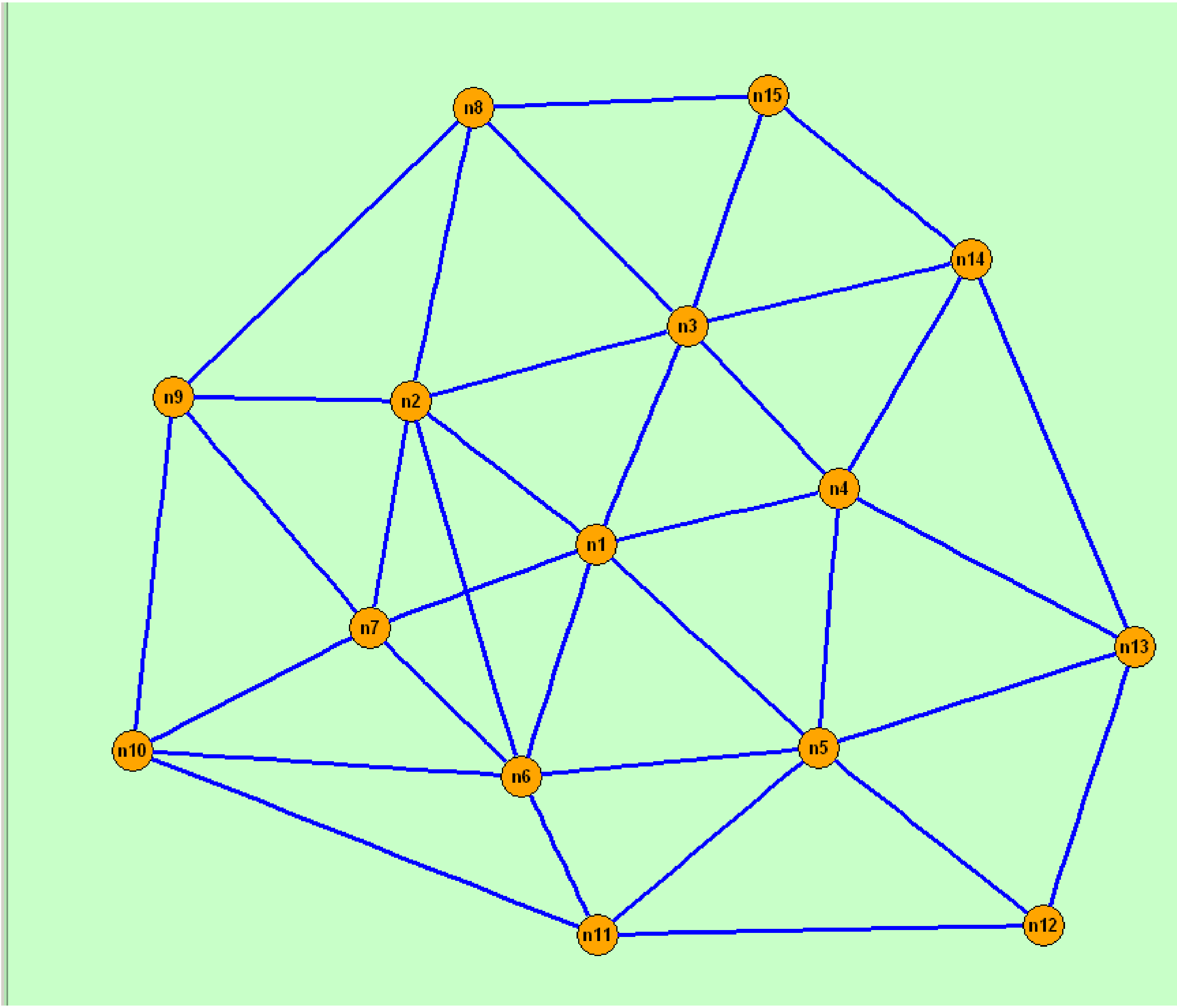

n1
n2
n3
n4
n5
n6
n7
n8
n9
n10
n11
n12
n13
n14
n15
n16
n17
n18
n19
n20
n21
n22
n23
n24
n25
n26
n27
n28
n29
n30
n31
n32
n33
n34
n35

## Approximation Algorithm

No simple characterization is known for graphs that are *k*-colourable if $k > 2$. In general, it is a hard problem to determine the chromatic number of an arbitrary graph. No algorithm is known that finds a colouring pattern using the fewest possible colours for every input graph. However, there are colouring algorithms that can be used to approximate the best colouring in the sense that they may sometimes use more colours than necessary. The following is an algorithm known as the ***largest-first algorithm:***

1. order the nodes by their degrees creating a sequence: $\boldsymbol{v} = v_{1,} v_{2,} \ldots$ and consider a sequence $\boldsymbol{c} = c_{1,} c_{2,} \ldots$ of colours
2. assign $c_1$ to $v_1$ and keep assigning $c_1$ to nodes in $\boldsymbol{v}$ non adjacent to $v_1$ and previously coloured nodes until you can
3. assign $c_2$ to the node $v_i$ with the greatest degree in $\boldsymbol{v}$ that has not been assigned a colour and then continue assigning $c_2$ to others nodes in $\boldsymbol{v}$ non-adjacent to $v_i$ until you can
4. continue this process until all the nodes have been assigned colours


n16
n17
n11
n6
n12
n1
n10
n2
n3
n7
n15
n4
n5
n13
n20
n9
n8
n18
n14
n19



n8
n15
n14
n3
n9
n2
n4
n1
n7
n13
n10
n6
n5
n11
n12