

برای هر نود یک مقدار  $height$  تعریف می‌کنیم. که نشان دهنده ارتفاع آن است.

پس تابعی غیر بازگشتی برای محاسبه ارتفاع هر نود تعریف می‌کنیم.

```
node:
  data
  left child
  right child
  height
```

calculate\_height(node root):

```
  curr
  queue nodes
  height ← 0
  node_count ← 0
  if (root = null)
    return 0
  nodes.enqueue(root)
  while (nodes.size > 0)
    height++
    node_count ← nodes.size
    while (node_count > 0)
      node_count--
      curr ← nodes.front
      if (curr->left ≠ null)
        nodes.enqueue(curr->left)
      if (curr->right ≠ null)
        nodes.enqueue(curr->right)
      nodes.dequeue()
```

این تابع در (اندازه زیردرخت  $x$ )  $O(5)$  مقدار ارتفاع نود  $x$  را محاسبه می‌کند.

حالا بر روی درخت پیاپی انجام داده و ارتفاع تمام نودها را محاسبه کنیم .  
پس بار دیگر بر روی درخت پیاپی انجام داده و برای هر نود چک می کنیم که ارتفاع دو نوزند  
آن حداکثر یک واحد تفاوت داشته باشد .

```
preorder (root)
    if (root == null)
        return
    stack nodes
    node-push (root)
    while (nodes.size() > 0)
        node ← nodes.top()
        nodes.pop
        node.height ← calculate-height (node)
        if (node → right)
            nodes.push (node → right)
        if (node → left)
            node.push (node → left)
```

بار دیگر preorder را انجام می دهیم اما این بار در هر نود بررسی می کنیم که ارتفاع  
زیردرخت چپ و راست حداکثر یک واحد اختلاف داشته باشند