

Aufgabe 36

Implementierung Um den Binärbaum als sortierte Folge in das Array zu übertragen muss man ihn ganz einfach mit Infix-Notation traversieren. Dazu gibt die Methode `binSort(BinTreeNode node, int[] a, int pos)` bei jedem Aufruf die Anzahl von schon gespeicherten Elementen zurück, um für nachfolgende Funktionsaufrufe die passende Speicheradresse zu bestimmen.

So sieht die von uns gewählte Implementierung aus:

```
public int binSort(BinTreeNode node, int[] a, int pos) {
    // Anzahl der schon eingefuegten Elemente
    int i = pos;

    // Linken Teilbaum traversieren
    if (node.getLeftChild() != null)
        i += binSort(node.getLeftChild(), a, i);

    // Element speichern
    a[i] = node.getValue();
    i++;

    // Rechten Teilbaum traversieren
    if (node.getRightChild() != null)
        i += binSort(node.getRightChild(), a, i);

    // Anzahl der eingefuegten Elemente zurueckgeben
    return i;
}
```

Laufzeit Die Laufzeit ist, da es sich um einen rekursiven Algorithmus handelt, abhängig von der Anzahl von Methodenaufrufen (da die restliche Laufzeit eines Methodenaufrufs konstant ist). Somit, da in jedem Fall für jeden im Baum vorhandenen Knoten die Methode aufgerufen wird, beträgt die Laufzeit dieses Algorithmus' $T(n) \in \Theta(n)$.