30. Mai 2016

Aufgabe 17

c)

Beh.: Ein Aufruf von `myAlgo(x, array, a, b)`, mit x vom Typ T, array vom Typ T[] und a,b ∈ ℕ mit 0 ≤ a < b < array.length, prüft, ob das Element x in { array[i] | a ≤ i < b } enthalten ist.

Bew.: Wenn man `myAlgo(x, array, a, b)` aufruft, mit x vom Typ T, array vom Typ T[] und a,b ∈ $\mathbb{N}_0$ mit 0 ≤ a < b < array.length, so gilt:

```
public static <T extends Comparable<T>> boolean myAlgo(T x, T[] array, int l, int r)
{
    if (r-l == 1)
    {
        // Position (3)
        return (array[1].compareTo(x)==0);
    }
    else
    {
        int m = (l+r)/2;
        return (
                // Position (1)
                myAlgo(x, array, l, m) ||
                // Position (2)
                myAlgo(x, array, m, r)
               );
    }
}
```

- An Position (1)

  $(l < r+1) \land x \notin \{ array[i] \mid a \le i < l \} \land (x \notin \{ array[i] \mid m \le i < b \} \lor x \in \{ array[i] \mid m \le i < b \})$

- An Position (2)

  $(l < r+1) \land x \notin \{ array[i] \mid a \le i < m \} \land (x \notin \{ array[i] \mid l \le i < b \} \lor x \in \{ array[i] \mid l \le i < b \})$

- An Position (3)

  $(l = r-1) \land x \notin \{ array[i] \mid a \le i < l \} \land (x = array[l] \lor x \neq array[l])$

Dies folgt auch aus dem Short-Curcuit-Operator im else-Zweig, da zuerst rekursiv $x \in \{ array[i] \mid a \le i < m \}$ überprüft wird, und nach einer erfolglosen Suche der Rest nach dem gleichen Prinzip untersucht wird.

Außerdem terminiert dieser Algorithmus, denn:
Entweder gelangt man in den if-Zweig, wenn r-l = 1, der sofortige Terminierung bedeutet.
Oder man gelangt in den else-Zweig, wo die Methode sich selbst aufruft, allerdings gilt: $l_1 > l_0$ oder

$r_1 < r_0$, aber trotzdem gilt weiterhin: $l_1 < r_1$. (l_1 und r_1 sind die Parameter des neuen Aufrufs, r_0 und l_0 die aktuellen Parameter). Dies führt zwangsläufig auch zur Terminierung.

Aus obigen Feststellungen folgt die Behauptung. □

a)

Nach c) überprüft ein Aufruf von `myAlgo(x, array, 0, array.length)`, ob das Element x im Array array enthalten ist.

b)

Die Operationen für das teilen und zusammenfügen sind konstant, d.h. f(n) = c. Es gilt nun: $T(n) = 2\,T(n/2) + c$. Es folgt nun mit Mastertheorem: $f(n) \in O(n^{1-\varepsilon})$ für $1 > \varepsilon > 0$.

Also gilt: $T(n) \in \Theta(n)$.