②

Binäre Suche

a) ges. Komplexität beider Varianten bestimmen mittels Master-Theorem

Lsg. Hat man ein Array der Größe N zu durchsuchen, teilt binary-search das Problem in den nicht-rekursiven Teil und das Problem ein Array der Größe $N/2$ zu durchsuchen, oder in der für das Master-Theorem gebräuchlichen mathematischen Schreibweise:

$$T(n) = T\left(\tfrac{n}{2}\right) + f(n) \Rightarrow \text{Rekursionsexponent } \varepsilon = \log_2 1 = 0$$

Fall 1: Referenzübergabe Ann. $f(n) \in \Theta(1)$

Aufwand geht mind. $f(n) \in \Omega(1)$

$\Rightarrow f(n) \in \Theta(1) = \Theta(n^0) \xrightarrow[\text{Master-Theorem}]{\text{Fall 2 stimmt}} T(n) \in \Theta(n^0 \log n)$

D.h. $T(n) \in \Theta(\log n)$ ✓

Fall 2: Kopieübergabe Ann. $f(n) \in O(n) \Leftrightarrow \forall n \geq n_0 : f(n) \leq c\, n$

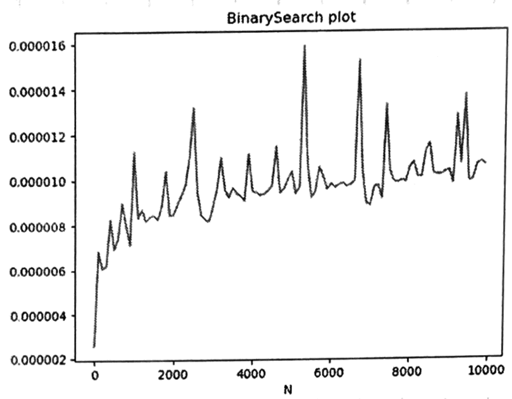
Leider lässt sich daraus keine weitere für das MT nützliche Information über eine untere Schranke ziehen.

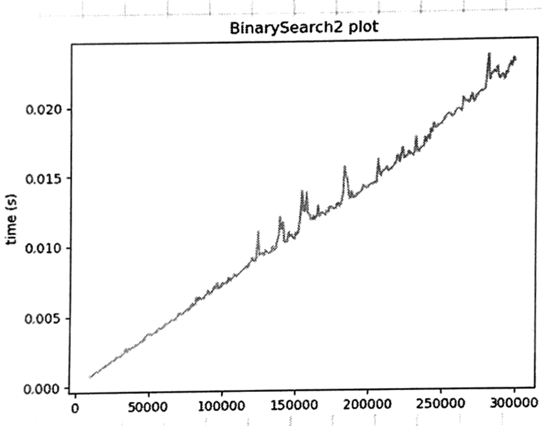
Zusätzl. Ann. $f(n) = n$

$\Rightarrow f(n) \in \Omega(n) = \Omega(n^{0+1}) \wedge a\, f\left(\tfrac{n}{b}\right) \leq c\, f(n)$ gilt für $c = 1/2$: $1 \cdot f\left(\tfrac{n}{2}\right) \leq \tfrac{n}{2}$ ✓

$\overset{\text{MT}}{\underset{\text{Fall 3}}{\Rightarrow}} T(n) \in \Theta(n)$ ✓

b) folgende Plots bestätigen diese Voraussage:





c)

14/14

A1 a) ✓

b) ✓

c) ✓

13/13