

DGA Ü3

1.) Sei S ein (unsortiertes) Feld von ganzen Zahlen und x eine ganze Zahl.

Gesucht ist ein Algorithmus, der feststellt ob für zwei Zahlen a, b aus S gilt $x = a + b$, mit worst-case Laufzeit von $\Theta(n \log n)$. $n = \text{len}(S)$

Idee: zuerst S sortieren, dann von links nach rechts die Zahlen durchgehen für den ersten Summanden a . Um b zu finden (oder eben nicht) mittels binary-search den rechts von a liegenden Teil des Feldes durchsuchen.

Algorithmus:

Aufwand:

is_sum_in_field(S, x):

$n = \text{length}(S);$

1

$S.\text{sort}();$

$n \cdot \log(n)$

for $a = 0, \dots, n$:

n

$l = a + 1; r = n - 1;$

n

while $l < r$:

$n \cdot \log_2(n - a)$

$m = \lfloor (l + r) / 2 \rfloor$

$n \cdot \log_2(n - a)$

if $(S[a] + S[m]) == x$:

$n \cdot \log_2(n - a)$

return true;

else if $(S[a] + S[m]) < x$:

$n \cdot \log_2(n - a)$

$l = m + 1;$

else:

$n \cdot \log_2(n - a)$

$r = m - 1;$

$n \cdot \log_2(n - a)$

return false;

1

\Rightarrow insgesamt Aufwand $\Theta(n \log(n))$