

# Binäre Suche

Wednesday, January 5, 2022 11:19 PM

## Binäre Suche in einem Array

Eine effizientere Suche als die lineare ist möglich, wenn das Array in sortierter Form vorliegt. Man wählt das Element in der Mitte des Arrays (ggf. abrunden) und entscheidet, ob das gesuchte Element damit gefunden wurde oder ob man davor oder dahinter weitersuchen muss. Dadurch wird der Suchbereich halbiert und man kann das Verfahren so lange fortsetzen, bis das gesuchte Element gefunden wurde, bzw. bis ausgeschlossen ist, dass sich das Element im Array befindet.

### Beispiel : Suche 8

1. Schritt	2	7	8	Mitte 12	13	19	23
2. Schritt	2	Mitte 7	8	12	13	19	23
3. Schritt	2	7	Mitte 8	12	13	19	23

Nach nur 3 Schritten ist eine binäre Suche in einem Array mit 7 Elementen spätestens beendet. Entweder das gesuchte Element wurde dann gefunden oder man kann sicher sein, dass es nicht in dem Array vorkommt.

### Bemerkung

Der Algorithmus der binären Suche arbeitet nach dem Prinzip Teile und Herrsche (Divide and Conquer).

Bei diesem Problemlösungsansatz wird das Ausgangsproblem so lange in kleinere Teilprobleme zerlegt, bis man diese lösen (beherrschen) kann. Danach bildet man aus diesen Teillösungen eine Lösung für das Gesamtproblem.

## Effizienzbetrachtung

Größe des Arrays	Anzahl Halbierungen	Anzahl Vergleiche
$2 = 2^1$	$1 = \log_2 2$	$1 + 1$
$4 = 2^2$	$2 = \log_2 4$	$2 + 1$
$8 = 2^3$	$3 = \log_2 8$	$3 + 1$
$16 = 2^4$	$4 = \log_2 16$	$4 + 1$
...	...	...
$1024 = 2^{10}$	$10 = \log_2 1024$	$10 + 1$
...	...	...
$n$	$\log_2 n$	$\log_2 n + 1$

## Laufzeit binäre Suche

best case:  $T(n) = 1$

worst case:  $T(n) = \log_2 n + 1$