

Binäre Suche

>> Bei der **Binären Suche** besteht die Idee des Algorithmus darin, nach dem Prinzip „Teile und herrsche“ vorzugehen, indem der Suchbereich jedes Mal halbiert wird, um ein gesuchtes Element zu finden.

Allerdings ist die Voraussetzung, dass unser Array bereits sortiert ist.

Pseudocode Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn sich das gesuchte Element in der Mitte befindet, bist du fertig.

Andernfalls:

wenn das gesuchte Element kleiner als das in der Mitte ist, wiederhole den Vorgang nachdem du den Endpunkt auf den Index gesetzt hast, der links von der Mitte ist.

oder, wenn das gesuchte Element größer als das in der Mitte ist, wiederhole den Vorgang nachdem du den Startpunkt auf den Index gesetzt hast, der rechts von der Mitte ist.

Beispiel

gesucht: 19

11 23 8 14 30 9 6 17 22 28 25 15 7 10 19

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht:

11	23	8	14	30	9	6	17	22	28	25	15	7	10	19
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht:

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 19

Start: 0

Ende: 14

Mitte: 7

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 19

Start: 0

Ende: 14

Mitte: 7

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 19

Start: 8

Ende: 14

Mitte: 7

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 19

Start: 8

Ende: 14

Mitte: 11

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht:

Start:

Ende:

Mitte:

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

**Gesuchtes Element kleiner als das in der Mitte?
Endpunkt = Index links von der Mitte. Weiter!**

Gesuchte Element größer als das in der Mitte?
Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 19

Start: 8

Ende: 10

Mitte: 11

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 19

Start: 8

Ende: 10

Mitte: 9

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 19

Start: 8

Ende: 10

Mitte: 9

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 16

Start: 0

Ende: 14

Mitte:

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 16

Start: 0

Ende: 14

Mitte: 7

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht:

Start:

Ende:

Mitte:

<input type="text" value="6"/>	<input type="text" value="7"/>	<input type="text" value="8"/>	<input type="text" value="9"/>	<input type="text" value="10"/>	<input type="text" value="11"/>	<input type="text" value="14"/>	<input type="text" value="15"/>	<input type="text" value="17"/>	<input type="text" value="19"/>	<input type="text" value="22"/>	<input type="text" value="23"/>	<input type="text" value="25"/>	<input type="text" value="28"/>	<input type="text" value="30"/>
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 16

Start: 8

Ende: 14

Mitte: 11

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 16

Start: 8

Ende: 10

Mitte: 11

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 16

Start: 8

Ende: 10

Mitte: 9

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

**Gesuchtes Element kleiner als das in der Mitte?
Endpunkt = Index links von der Mitte. Weiter!**

Gesuchte Element größer als das in der Mitte?
Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel gesucht: Start: Ende: Mitte:

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

**Gesuchtes Element kleiner als das in der Mitte?
Endpunkt = Index links von der Mitte. Weiter!**

Gesuchte Element größer als das in der Mitte?
Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 16

Start: 8

Ende: 8

Mitte: 8



Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht:

Start:

Ende:

Mitte:

<input type="text" value="6"/>	<input type="text" value="7"/>	<input type="text" value="8"/>	<input type="text" value="9"/>	<input type="text" value="10"/>	<input type="text" value="11"/>	<input type="text" value="14"/>	<input type="text" value="15"/>	<input type="text" value="17"/>	<input type="text" value="19"/>	<input type="text" value="22"/>	<input type="text" value="23"/>	<input type="text" value="25"/>	<input type="text" value="28"/>	<input type="text" value="30"/>
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

**Gesuchtes Element kleiner als das in der Mitte?
Endpunkt = Index links von der Mitte. Weiter!**

Gesuchte Element größer als das in der Mitte?
Startpunkt = Index rechts von der Mitte. Weiter!

Beispiel

gesucht: 16

Start: 8

Ende: 7

Mitte: 8

6	7	8	9	10	11	14	15	17	19	22	23	25	28	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wiederhole, bis das (Teil-)Array die Größe 0 hat:

Berechne den Mittelpunkt des aktuellen (Teil-)Arrays.

Wenn gesuchtes Element in der Mitte ist: fertig!

Andernfalls:

Gesuchtes Element kleiner als das in der Mitte?

Endpunkt = Index links von der Mitte. Weiter!

Gesuchte Element größer als das in der Mitte?

Startpunkt = Index rechts von der Mitte. Weiter!

Fälle Worst Case:

Wir müssen die n Elemente „bis zum Ende“ immer weiter halbieren, um das gesuchte Element zu finden – weil das Zielement entweder erst am Ende der letzten Teilung gefunden wird oder gar nicht vorhanden ist.

Best Case:

Das gesuchte Element befindet sich in der Mitte und wird sofort, also beim ersten Vergleich, gefunden.

**Laufzeit-
komplexität** $O(\log n)$
 $\Omega(1)$

EXTRAS IN 3 MINUTEN
FRAGEN – ANTWORTEN – RÄTSEL
UND KURZE ZUSAMMENFASSUNG

F1: Was nimmt man als Mitte, wenn die Elementzahl im Array gerade ist?
Zum Beispiel bei zwölf Elementen?

F2: Wie viele Vergleiche sind bei der binären Suche maximal nötig, um einen Wert in einem Array mit 20 Elementen zu suchen?

A2: 5 Vergleiche

weil $\log_2(20) = 4,32$, d.h. aufgerundet 5

Gedankenspiel: „20 Questions“ mit binärer Suche

Ziel bei „20 Questions“: Durch geschickte Ja-Nein-Fragen ein Objekt oder eine Person erraten, das bzw. die sich jemand anderes ausgedacht hat.

Effektive Strategie: mit jeder Frage die noch in Frage kommende Antwortmöglichkeiten halbieren

Mit den *richtigen* 20 Fragen könnte man dann aus einer Menge von mehr als einer Million Möglichkeiten das Gesuchte ermitteln.

Um den ersten Buchstaben eines Wortes zu ermitteln (Alphabet: 26 Buchstaben) genügen max. fünf Fragen. *Warum?*

Die übrigen 15 Fragen reichen dann aber evtl. nur noch, um drei weitere Buchstaben zu ermitteln.

Optimierung der Strategie

Ideen: Häufige Buchstabenkombinationen nutzen (z. B. "S-C-H") und nach Anfangs- und Endbuchstaben fragen.

Hoffnung: z.B. erste zwei Buchstaben "S-C"
→ dritte Frage nicht mehr nötig!

Weitere Optimierung durch Ausnutzen sprachlicher Eigenheiten.

Anfangsbuchstaben "S", "A", "E", "K", "B":
50% aller deutschen Wörter

Zuerst nach diesen fragen, um den Suchraum schneller einzugrenzen.

Prinzip: Teile und Herrsche
Wiederholte Halbierung
des Suchbereichs

Best-Case Komplexität:
 $\Omega(1)$
Element wird direct
in der Mitte gefunden

Voraussetzung:
Sortiertes Array

Abbruch bei erfolgloser
Suche: Wenn sich Start-
und Endindex überkreuzen.

Worst-Case Komplexität:
 $O(\log n)$
Deutlich besser als
 $O(n)$ bei linearer Suche

Praktische Anwendung:
Effizient bei großen,
sortierten Datensätzen