

Wilhelm Buchmüller 3133783
 Daniel Waner 3149308
 Artur Frenzen 2736424
 Aufgabe 1

a)

Lineare Suche:

Zeiger	■														
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger		■													
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger			■												
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger				■											
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger					■										
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger						■									
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger							■								
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger								■							
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger									■						
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger										■					
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Return this.Array(9)

Wilhelm Buchmüller 3133783
 Daniel Waner 3149308
 Artur Frenzen 2736424
 Binäre Suche:

Das Array ist sortiert also kann mit der Suche begonnen werden:

Wählen eines Mittelpunktes(48)

Zeiger	<div style="background-color: #cccccc; width: 100%; height: 15px; position: relative;"><div style="position: absolute; left: 48%; width: 2px; height: 10px; background-color: black;"></div></div>														
Wert	12	20	29	33	38	42	47	48	49	52	62	68	71	84	92
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wert ist größer als 48 also kann die linke Hälfte UND 48 “vergessen” werden.

Wählen eines neuen Mittelpunktes(62)

Zeiger	<div style="background-color: #cccccc; width: 100%; height: 15px; position: relative;"><div style="position: absolute; left: 28%; width: 2px; height: 10px; background-color: black;"></div></div>														
Wert	49	52	62	68	71	84	92								
Index	8	9	10	11	12	13	14								

Wert ist kleiner als 62 also wird weiter auf der linken Seite gesucht

Wählen eines neuen Mittelpunktes(49)

Zeiger	<div style="background-color: #cccccc; width: 100%; height: 15px; position: relative;"><div style="position: absolute; left: 26%; width: 2px; height: 10px; background-color: black;"></div></div>														
Wert		49				52									
Index		8				9									

Wert ist größer als 49, wählen eines Mittelpunktes(52)

Wert ist (52) an Index 9

return

b)

Lineare Suche:

Zeiger	■														
Wert	3	23	28	30	33	39	38	41	44	48	49	59	73	81	96
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger		■													
Wert	3	23	28	30	33	39	38	41	44	48	49	59	73	81	96
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger			■												
Wert	3	23	28	30	33	39	38	41	44	48	49	59	73	81	96
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Wilhelm Buchmüller 3133783

Daniel Waner 3149308

Artur Frenzen 2736424

Zeiger				■											
Wert	3	23	28	30	33	39	38	41	44	48	49	59	73	81	96
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger					■										
Wert	3	23	28	30	33	39	38	41	44	48	49	59	73	81	96
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger						■									
Wert	3	23	28	30	33	39	38	41	44	48	49	59	73	81	96
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Zeiger							■								
Wert	3	23	28	30	33	39	38	41	44	48	49	59	73	81	96
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Return this.Array(6)

Binäre Suche:

Ist nicht möglich da das Array nicht sortiert ist.

Aufgabe 4

Fall 1: InsertionSort & BubbleSort. In Beiden Fällen beträgt die Komplexität $O(n)$.

Fall 2: MergeSort, da im Worst Case die Komplexität $O(n \log(n))$ beträgt

Fall 3: Wieder MergeSort, da die Komplexität maximal $O(n \log(n))$ beträgt. QuickSort allerdings unter Umständen auch, meist $O(n \log(n))$, schlechtestens $O(n^2)$, dies tritt jedoch selten auf.

-