# ÜBUNGSBLATT 1

## Team:

Datenstrukturen & Algorithmen

Sommersemester 2017

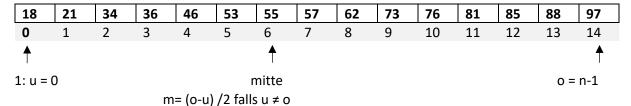
Hasan Darwish (3247569),

Polina Jungblut (3254837)

Ina Vasileiadou (3124938)

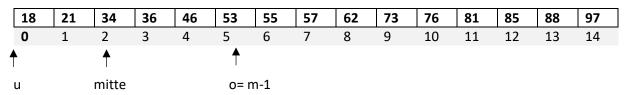
#### Aufgabe 1:

- a) Binäre Suche:
- (1). Schlüssel: k = 53; Länge: n;



vergleich: (Wenn m = k -> fertig) hier nicht der Fall

(2) Wenn m > k wird die obere Schranke zu o = m-1 und es wird rekursiv in der (ersten)Hälfte in der sich das Element befindet weiter verglichen (hier der Fall)



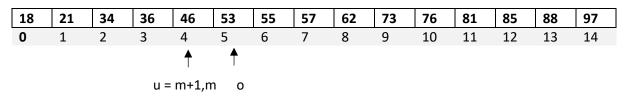
vergleich: (Wenn m = k -> fertig) hier nicht der Fall

(3) Wenn m < k wird: u = m+1, das Element befindet sich in der hinteren Hälfte der Folge und nur diese Elemente werden mit dem Schlüssel verglichen

18	21	34	36	46	53	55	57	62	73	76	81	85	88	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
			$\uparrow$	<b>†</b>	$\uparrow$									
		u = 1	m+1	m	0									

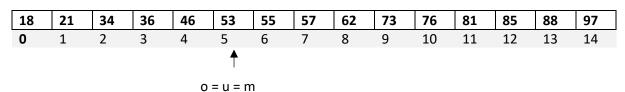
vergleich: (Wenn m = k -> fertig) hier nicht der Fall

(4) Wenn m < k wird: u = m+1, das Element befindet sich in der hinteren Hälfte der Folge und nur diese Elemente werden mit dem Schlüssel verglichen



vergleich: (Wenn m = k -> fertig) hier nicht der Fall

(5) Wenn m < k wird: u = m+1, das Element befindet sich in der hinteren Hälfte der Folge und nur diese Elemente werden mit dem Schlüssel verglichen



→ vergleich: (Wenn m = k -> fertig)

### Sequentielle Suche:

Schlüssel: k = 53; Länge: n;

18	21	34	36	46	53	55	57	62	73	76	81	85	88	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.														

Vergleich mit k F [0] ≠ k also weiter:

18	21	34	36	46	53	55	57	62	73	76	81	85	88	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-	2.													

Vergleich mit k F [1] ≠ k also weiter:

18	21	34	36	46	53	55	57	62	73	76	81	85	88	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	-	3.												

Vergleich mit k F [2] ≠ k also weiter:

18	21	34	36	46	53	55	57	62	73	76	81	85	88	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		1	4.											

Vergleich mit k F [3] ≠ k also weiter:

18	21	34	36	46	53	55	57	62	73	76	81	85	88	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
			5	· <b>/</b>										

Vergleich mit k F [4] ≠ k also weiter:

18	21	34	36	46	53	55	57	62	73	76	81	85	88	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
				<b>1</b> 6	•									

Vergleich mit k: F [5] = k;

7. Ende des Suchverfahrens;

8. Ausgabe: F [5] = 53

b)

Die *Binäre Suche* lässt sich hier nicht anwenden, da die Folge nicht sortiert ist, was eine Voraussetzung für die Binäre Suche ist. Somit kann man den Wahlfreien Zugriff von sortierten Folgen nicht ausnutzen und deshalb diese Form der Suche nicht anwenden.

Datenstrukturen & Algorithmen Sommersemester 2017 Übungsblatt 1 Hasan Darwish ,3247569 Polina Jungblut,3254837 Ina Vasileiadou, 3124938

## Sequentielle Suche:

Schlüssel: k = 64; Länge: n;

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.														

Vergleich mit k F [0] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-	2.													

Vergleich mit k F [1] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	-	3.												

Vergleich mit k F [2] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Ţ.	4.											

Vergleich mit k F [3] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
			5	•										

Vergleich mit k F [4] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
				<b>1</b> 6	<b>1</b>									

Vergleich mit k F [5] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
					<b>1</b> 7.	•								

Vergleich mit k F [6] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
						<b>8</b> .	1							

Vergleich mit k F [7] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	■ 9 ▲													

Vergleich mit k F [8] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
								<b>1</b> 0	•					

Vergleich mit k F [9] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
									<b>1</b> 11	l🎓				

Vergleich mit k F [10] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
										<b>1</b>	24			

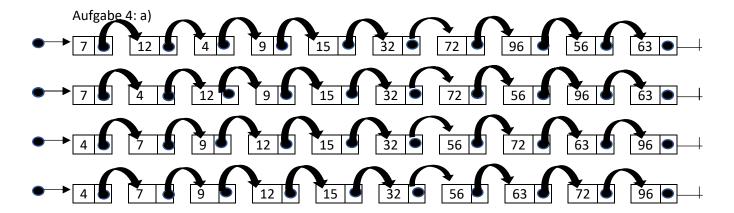
Vergleich mit k F [11] ≠ k also weiter:

12	15	21	22	24	31	40	48	55	59	71	88	64	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
											-1	2 🛕		

Vergleich mit k: F [12] = k;

14.Ende des Suchverfahrens;

15: Ausgabe: F [12] = 64



b) Bei der gezeigten Listenstruktur handelt es sich um eine einfach verkettete Liste (nur Zeiger auf Nachfolgeknoten), die eine Navigation von vorne nach hinten ermöglicht, aber nicht rückwärts wie bei einer doppelt verketteten Liste, somit kann der Bubblesort Algorithmus die Liste von vorne nach hinten abarbeiten und sortieren.