



ÜBUNGSBLATT 1

Team:

Datenstrukturen & Algorithmen

Sommersemester 2017

Hasan Darwish (3247569),

Polina Jungblut (3254837)

Ina Vasileiadou (3124938)

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 $k \neq F[0]$ 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 $k \neq F[1]$ 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 $k \neq F[2]$ 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

4. 

Vergleich mit $k \neq F[3]$ 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. 

Vergleich mit $k \neq F[4]$ 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

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.

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 $k \neq F[0]$ 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 $k \neq F[1]$ 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 $k \neq F[2]$ 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

4. 

Vergleich mit $k \neq F[3]$ 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 $k \neq F[4]$ 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

6. 

Vergleich mit $k \neq F[5]$ 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

7. 

Vergleich mit $k \neq F[6]$ 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

8. 


Vergleich mit $k \neq F[7]$ 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 $k \neq F[8]$ 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




Vergleich mit $k \neq F[9]$ 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




Vergleich mit $k \neq F[10]$ 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



Vergleich mit $k \neq F[11]$ 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

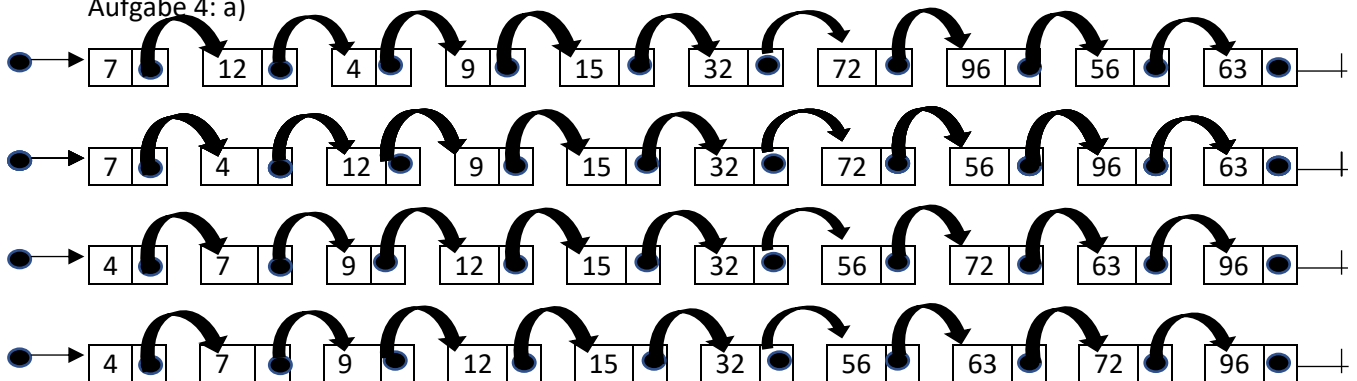


Vergleich mit k : $F[12] = k$;

14. Ende des Suchverfahrens;

15: Ausgabe: $F[12] = 64$

Aufgabe 4: a)



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.