DHBW Lörrach
Duale Hochschule
Baden-Württemberg

Algorithmen und Komplexität
TIF 21A/B
Dr. Bruno Becker

Übungsblatt 2: Analyse und Sortieren

Übungsblatt 2 –Aufgabe 1

a. Erstellen Sie rekursiven Algorithmus für Bubblesort.
b. Wann ist Bubblesort besonders günstig, wann besonders ungünstig?
c. Ist Bubblesort stabil?

```
Public void  bubbleSort(int[] a, int n)
{    // a array der Länge n
     swapped? = false
     for (int i=0; i < n-1, i++)
        {
         if (a[i] > a [i+1] )
            {
            swap (a, i,i+1); // swap a[i] with a [i+1]
            swapped? = true
            }
        }
     If (n-1 > 1) and (swapped?=true)
     {
        bubbleSort (a,n-1);  //In Rekursionstiefe k sind
       die letzten k Elemente schon sortiert
     }
}
```

Beispiel: 15,2,43,17,4,8

DL: 15,2 (Tausch in 2,15) 15,43,17 (tausch in 17,43) 43,4 (Tausch in 4,43) 43,8 (Tausch in 8,43)

Folge nach 1. DL: **2,15,17,4,8,43**

DL: 2,15,17,4 (Tausch in 4,17), 17,8 (Tausch in 8,17) 17,43

Folge nach 2. DL: **2,15,4,8,17,43**

DL 2,15,4 (Tausch in 4,15), 15,8 (Tausch in 8,15),15,17,43

Folge nach 3. DL: **2,4,8,15,17,43**

=> Folge sortiert

b) **Aufwand:** Best case (Folge sortiert) C(N) = N-1, M(N) = 0

Worst Case (Folge absteigend sortiert) $\Theta(N^2)$, auch für average case..

Bubblesort nur gut, wenn Folge „fast“ sortiert

c) Ist stabil (Nachbarn werden nur vertauscht bei echt >) und in-place

Übungsblatt 2 –Aufgabe 2

a) $f(n)= n^2$ $g(n) = n \log n$ => $f(n) \varepsilon \Omega (g(n))$, g ist untere Schranke für f

b) $f(n)= \sqrt{n}$ $g(n) = 500\ n$ => $f(n) \varepsilon O(g(n))$, g ist obere Schranke für f

c) $f(n) = 3 \log n$ $g(n) = \ln n$ => $f(n) \varepsilon \Theta(g(n))$, g ist obere und untere Schranke für f (wg. Logarithmus-Gesetzen)

d) $f(n) = 47\ n^2 – 12\ n + 18$ $g(n) = n^2$ $f(n) \varepsilon \Theta(g(n))$. z.B. für $f(n) \varepsilon O(g(n))$ z. B. c=47, n_o=2

Übungsblatt 2 –Aufgabe 3

a) Sortieren Sie die Folge mittels Insertion Sort. Geben Sie die Folge nach jedem Schleifendurchlauf an.
b) In welchem Fall benötigt Insertion Sort nur O(n) Vergleiche?

8	2	1	5	4	6	9	3	7

1. DL 2 8 1 5 4 6 9 3 7
2. DL: 1 2 8 5 4 6 9 3 7
3. DL: 1 2 5 8 4 6 9 3 7
4. DL: 1 2 4 5 8 6 9 3 7
5. DL: 1 2 4 5 6 8 9 3 7
6. DL: 1 2 4 5 6 8 9 3 7
7. DL: 1 2 3 4 5 6 8 9 7
8. DL: 1 2 3 4 5 6 7 8 9

b) Wenn Folge sortiert nur O (n) Vergleiche

Übungsblatt 2 –Aufgabe 4

a) Erläutern Sie die Funktionsweise von Quicksort

b) Gegeben sei die folgende Zahlenfolge

11	7	23	17	15	8

Rekursionstiefe sei k, S_k die Folge in der k-ten Rekursion, x_k das Pivot-Element der k-ten Rekursion

k	S_k	Pivot x_k	Kommentar
0	S_0 = (11, 7, 23, 17, 15, 8) x_0 i-> <-j	x_0 = 11	Start mit x_0 = 11
	S_0 = (11, 7, 23, 17, 15, 8) x_0 i j		i < j: vertausche a[i] mit a[j], fahre mit Suche fort
	S_0 = (11, 7, 8, 17, 15, 23) x_0 j i		i ≥ j; vertausche a[j] mit Pivot und setze mit Teilfolgen fort
	S_0 = (8, 7, 11, 17, 15, 23)= S_1 x_0 S_2		
1	S_1 = (8, 7) x_1 i,j	x_1 = 8	
	S_1 = (8, 7) x_1 i,j		i ≥ j; vertausche a[j] mit Pivot und setze mit Teilfolgen fort
	S_1 = (7, 8) = S_3 x_1		
2	S_2 = (17, 15, 23) x_2 i-> <-j	x_2 = 17	
	S_2 = (17, 15, 23) x_2 j i		i ≥ j; vertausche a[j] mit Pivot und setze mit Teilfolgen fort
	S_2 = (15, 17, 23)= S_4 x_2 S_5		
3	S_3 = (7)		Folge 1 Element-> Abbruch
4	S_3 = (15)		Folge 1 Element-> Abbruch
5	S_3 = (23)		Folge 1 Element-> Abbruch

Übungsblatt 2 – Aufgabe 5

Geben Sie für die unten angegebenen Zahlenfolgen jeweils die Laufzeit der Sortierverfahren Selection Sort, Insertion Sort und Bubblesort in O-Notation an (mit Begründung).

a) 1, $\frac{N}{2}$ +1, 2, $\frac{N}{2}$+2, …, $\frac{N}{2}$, N (N gerade)

b) N, 1, N-1, 2, N-2,3,…, N - $\frac{N}{2}$ +1, $\frac{N}{2}$ (N gerade)

c) N, 1,2,3,…,N-1

d) 2,3,4,…N,1

Selection Sort: $O(N^2)$ für a)-d), weil unabhängig von Sortierreihenfolge

Insertion Sort:

a) Bei jedem zweiten Element findet Swap statt, suche an die richtige Stelle zum Einfügen abhängig von Teilfolge => $O(N^2)$

b) Analog zu a) $O(N^2)$

c) O(N), Es muss zwar bei jedem Schleifen-Durchgang eingefügt werden, aber Einfügestelle immer an 2ter Stelle => O(N)

d) Nur 1x muss Swap durchgeführt werden => O(N)

Bubblesort:

a) Durch lokale Vertauschungen wandern Elemente langsam an richtige Stelle, z.B. 2tes Element N/2+ 1 benötigt ca N/2 Durchläufe => $O(N^2)$

b) Durch lokale Vertauschungen wandern Elemente langsam an richtige Stelle, z.B. letztes Element N/2 benötigt ca N/2 Durchläufe => $O(N^2)$

c) Maximum wandert in einem Durchlauf nach hinten => O(N)

d) Die 1 wandert nur sehr langsam nach vorne, pro Schleifendurchgang um 1 Position => $O(N^2)$