

Einzelprüfung „Theoretische Informatik / Algorithmen (vertieft)“

Einzelprüfungsnummer 66115 / 2021 / Frühjahr

Thema 1 / Teilaufgabe 2 / Aufgabe 1

(Sortieren)

Stichwörter: Sortieralgorithmen, Insertionsort, Quicksort, Mergesort

- (a) Geben Sie für folgende Sortierv Verfahren jeweils zwei Felder A und B an, so dass das jeweilige Sortierv Verfahren angewendet auf A seine Best-Case-Laufzeit und angewendet auf B seine Worst-Case-Laufzeit erreicht. (Wir messen die Laufzeit durch die Anzahl der Vergleiche zwischen Elementen der Eingabe.) Dabei soll das Feld A die Zahlen $1, 2, \dots, 7$ genau einmal enthalten; das Feld B ebenso. Sie bestimmen also nur die Reihenfolge der Zahlen.

Wenden Sie als Beleg für Ihre Aussagen das jeweilige Sortierv Verfahren auf die Felder A und B an und geben Sie nach jedem größeren Schritt des Algorithmus den Inhalt der Felder an.

Geben Sie außerdem für jedes Verfahren asymptotische Best- und Worst-Case-Laufzeit für ein Feld der Länge n an.

Die im Pseudocode verwendete Unter routine $\text{Swap}(A, i, j)$ vertauscht im Feld A die jeweiligen Elemente mit den Indizes i und j miteinander.

- (i) **Insertionsort**

Best-Case

1	2	3	4	5	6	7
---	---	---	---	---	---	---

```
1  2  3  4  5  6  7  Eingabe
1  2* 3  4  5  6  7  markiere (i 1)
1  2  3* 4  5  6  7  markiere (i 2)
1  2  3  4* 5  6  7  markiere (i 3)
1  2  3  4  5* 6  7  markiere (i 4)
1  2  3  4  5  6* 7  markiere (i 5)
1  2  3  4  5  6  7* markiere (i 6)
1  2  3  4  5  6  7  Ausgabe
```

Worst-Case

7	6	5	4	3	2	1
---	---	---	---	---	---	---

```
7  6  5  4  3  2  1  Eingabe
7  6* 5  4  3  2  1  markiere (i 1)
>7 7< 5  4  3  2  1  vertausche (i 0<>1)
6  7  5* 4  3  2  1  markiere (i 2)
6 >7 7< 4  3  2  1  vertausche (i 1<>2)
```

```

>6 6< 7 4 3 2 1 vertausche (i 0<>1)
5 6 7 4* 3 2 1 markiere (i 3)
5 6 >7 7< 3 2 1 vertausche (i 2<>3)
5 >6 6< 7 3 2 1 vertausche (i 1<>2)
>5 5< 6 7 3 2 1 vertausche (i 0<>1)
4 5 6 7 3* 2 1 markiere (i 4)
4 5 6 >7 7< 2 1 vertausche (i 3<>4)
4 5 >6 6< 7 2 1 vertausche (i 2<>3)
4 >5 5< 6 7 2 1 vertausche (i 1<>2)
>4 4< 5 6 7 2 1 vertausche (i 0<>1)
3 4 5 6 7 2* 1 markiere (i 5)
3 4 5 6 >7 7< 1 vertausche (i 4<>5)
3 4 5 >6 6< 7 1 vertausche (i 3<>4)
3 4 >5 5< 6 7 1 vertausche (i 2<>3)
3 >4 4< 5 6 7 1 vertausche (i 1<>2)
>3 3< 4 5 6 7 1 vertausche (i 0<>1)
2 3 4 5 6 7 1* markiere (i 6)
2 3 4 5 6 >7 7< vertausche (i 5<>6)
2 3 4 5 >6 6< 7 vertausche (i 4<>5)
2 3 4 >5 5< 6 7 vertausche (i 3<>4)
2 3 >4 4< 5 6 7 vertausche (i 2<>3)
2 >3 3< 4 5 6 7 vertausche (i 1<>2)
>2 2< 3 4 5 6 7 vertausche (i 0<>1)
1 2 3 4 5 6 7 Ausgabe

```

- (ii) Standardversion von **Quicksort** (Pseudocode s.u., Feldindizes beginnen bei 1), bei der das letzte Element eines Teilfeldes als Pivot-Element gewählt wird.

Funktion Quicksort($A, l = 1, r = A.length$)

```

if  $l < r$  then
     $m = \text{Partition}(A, l, r);$ 
    Quicksort( $A, l, m - 1$ );
    Quicksort( $A, m + 1, r$ );
end

```

Funktion Partition(A, int l, int r)

```
pivot = A[r];  
i = l;  
for j = l to r - 1 do  
    if A[j] < pivot then  
        Swap(A, i, j);  
        i = i + 1;  
    end  
end  
end
```

Lösungsvorschlag

Best-Case

1	3	2	6	5	7	4
---	---	---	---	---	---	---

```
1  3  2  6  5  7  4  zerlege  
1  3  2  6  5  7  4* markiere (i 6)  
>1< 3  2  6  5  7  4  vertausche (i 0<>0)  
1  >3< 2  6  5  7  4  vertausche (i 1<>1)  
1  3  >2< 6  5  7  4  vertausche (i 2<>2)
```

```

1 3 2 >6 5 7 4< vertausche (i 3<>6)
1 3 2 zerlege
1 3 2* markiere (i 2)
>1< 3 2 vertausche (i 0<>0)
1 >3 2< vertausche (i 1<>2)
      5 7 6 zerlege
      5 7 6* markiere (i 6)
      >5< 7 6 vertausche (i 4<>4)
      5 >7 6< vertausche (i 5<>6)

```

Worst-Case

7	6	5	4	3	2	1
---	---	---	---	---	---	---

```

1 2 3 4 5 6 7 zerlege
1 2 3 4 5 6 7* markiere (i 6)
>1< 2 3 4 5 6 7 vertausche (i 0<>0)
1 >2< 3 4 5 6 7 vertausche (i 1<>1)
1 2 >3< 4 5 6 7 vertausche (i 2<>2)
1 2 3 >4< 5 6 7 vertausche (i 3<>3)
1 2 3 4 >5< 6 7 vertausche (i 4<>4)
1 2 3 4 5 >6< 7 vertausche (i 5<>5)
1 2 3 4 5 6 >7< vertausche (i 6<>6)
1 2 3 4 5 6 zerlege
1 2 3 4 5 6* markiere (i 5)
>1< 2 3 4 5 6 vertausche (i 0<>0)
1 >2< 3 4 5 6 vertausche (i 1<>1)
1 2 >3< 4 5 6 vertausche (i 2<>2)
1 2 3 >4< 5 6 vertausche (i 3<>3)
1 2 3 4 >5< 6 vertausche (i 4<>4)
1 2 3 4 5 >6< vertausche (i 5<>5)
1 2 3 4 5 zerlege
1 2 3 4 5* markiere (i 4)
>1< 2 3 4 5 vertausche (i 0<>0)
1 >2< 3 4 5 vertausche (i 1<>1)
1 2 >3< 4 5 vertausche (i 2<>2)
1 2 3 >4< 5 vertausche (i 3<>3)
1 2 3 4 >5< vertausche (i 4<>4)
1 2 3 4 zerlege
1 2 3 4* markiere (i 3)
>1< 2 3 4 vertausche (i 0<>0)
1 >2< 3 4 vertausche (i 1<>1)
1 2 >3< 4 vertausche (i 2<>2)
1 2 3 >4< vertausche (i 3<>3)
1 2 3 zerlege

```

1 2 3*	markiere (i 2)
>1< 2 3	vertausche (i 0<>0)
1 >2< 3	vertausche (i 1<>1)
1 2 >3<	vertausche (i 2<>2)
1 2	zerlege
1 2*	markiere (i 1)
>1< 2	vertausche (i 0<>0)
1 >2<	vertausche (i 1<>1)

- (iii) **QuicksortVar**: Variante von Quicksort, bei der immer das mittlere Element eines Teilfeldes als Pivot-Element gewählt wird (Pseudocode s.u., nur eine Zeile neu). Bei einem Aufruf von PartitionVar auf ein Teilfeld $A[l \dots r]$ wird also erst mithilfe der Unterroutine Swap $A \left[\left\lfloor \frac{l+r-1}{2} \right\rfloor \right]$ mit $A[r]$ vertauscht.

Funktion QuicksortVar($A, l = 1, r = A.length$)

```

if  $l < r$  then
     $m = \text{PartitionVar}(A, l, r);$ 
    QuicksortVar( $A, l, m - 1$ );
    QuicksortVar( $A, m + 1, r$ );
end

```

Funktion PartitionVar($A, \text{int } l, \text{int } r$)

```

Swap( $A, \left\lfloor \frac{l+r-1}{2} \right\rfloor, r$ );
pivot =  $A[r]$ ;
 $i = l$ ;
for  $j = l$  to  $r - 1$  do
    if  $A[j] < \text{pivot}$  then
        Swap( $A, i, j$ );
         $i = i + 1$ ;
    end
end

```

Best-Case

1	2	3	4	5	6	7
---	---	---	---	---	---	---

```
1  2  3  4  5  6  7  zerlege
1  2  3  4* 5  6  7  markiere (i 3)
1  2  3 >4  5  6  7< vertausche (i 3<>6)
>1< 2  3  7  5  6  4  vertausche (i 0<>0)
1 >2< 3  7  5  6  4  vertausche (i 1<>1)
1  2 >3< 7  5  6  4  vertausche (i 2<>2)
```

```

1 2 3 >7 5 6 4< vertausche (i 3<>6)
1 2 3 zerlege
1 2* 3 markiere (i 1)
1 >2 3< vertausche (i 1<>2)
>1< 3 2 vertausche (i 0<>0)
1 >3 2< vertausche (i 1<>2)
      5 6 7 zerlege
      5 6* 7 markiere (i 5)
      5 >6 7< vertausche (i 5<>6)
    >5< 7 6 vertausche (i 4<>4)
      5 >7 6< vertausche (i 5<>6)
1 2 3 4 5 6 7 Ausgabe

```

Worst-Case

2	4	6	7	1	5	3
---	---	---	---	---	---	---

```

2 4 6 7 1 5 3 zerlege
2 4 6 7* 1 5 3 markiere (i 3)
2 4 6 >7 1 5 3< vertausche (i 3<>6)
>2< 4 6 3 1 5 7 vertausche (i 0<>0)
2 >4< 6 3 1 5 7 vertausche (i 1<>1)
2 4 >6< 3 1 5 7 vertausche (i 2<>2)
2 4 6 >3< 1 5 7 vertausche (i 3<>3)
2 4 6 3 >1< 5 7 vertausche (i 4<>4)
2 4 6 3 1 >5< 7 vertausche (i 5<>5)
2 4 6 3 1 5 >7< vertausche (i 6<>6)
2 4 6 3 1 5 zerlege
2 4 6* 3 1 5 markiere (i 2)
2 4 >6 3 1 5< vertausche (i 2<>5)
>2< 4 5 3 1 6 vertausche (i 0<>0)
2 >4< 5 3 1 6 vertausche (i 1<>1)
2 4 >5< 3 1 6 vertausche (i 2<>2)
2 4 5 >3< 1 6 vertausche (i 3<>3)
2 4 5 3 >1< 6 vertausche (i 4<>4)
2 4 5 3 1 >6< vertausche (i 5<>5)
2 4 5 3 1 zerlege
2 4 5* 3 1 markiere (i 2)
2 4 >5 3 1< vertausche (i 2<>4)
>2< 4 1 3 5 vertausche (i 0<>0)
2 >4< 1 3 5 vertausche (i 1<>1)
2 4 >1< 3 5 vertausche (i 2<>2)
2 4 1 >3< 5 vertausche (i 3<>3)
2 4 1 3 >5< vertausche (i 4<>4)
2 4 1 3 zerlege

```

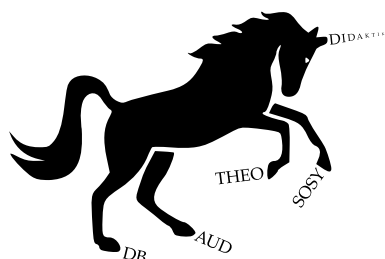

2 4* 1 3	markiere (i 1)
2 >4 1 3<	vertausche (i 1<>3)
>2< 3 1 4	vertausche (i 0<>0)
2 >3< 1 4	vertausche (i 1<>1)
2 3 >1< 4	vertausche (i 2<>2)
2 3 1 >4<	vertausche (i 3<>3)
2 3 1	zerlege
2 3* 1	markiere (i 1)
2 >3 1<	vertausche (i 1<>2)
>2< 1 3	vertausche (i 0<>0)
2 >1< 3	vertausche (i 1<>1)
2 1 >3<	vertausche (i 2<>2)
2 1	zerlege
2* 1	markiere (i 0)
>2 1<	vertausche (i 0<>1)
>1< 2	vertausche (i 0<>0)
1 >2<	vertausche (i 1<>1)

(b) Geben Sie die asymptotische Best- und Worst-Case-Laufzeit von **Mergesort** an.

Lösungsvorschlag

Best-Case: $\mathcal{O}(n \cdot \log(n))$

Worst-Case: $\mathcal{O}(n^2)$



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/66115/2021/03/Thema-1/Teilaufgabe-2/Aufgabe-1.tex>