

Aufgabe 5 (Sortierverfahren)

In der folgenden Aufgabe soll ein Feld A von ganzen Zahlen aufsteigend sortiert werden. Das Feld habe n Elemente A[0] bis A[n-1]. Der folgende Algorithmus (in der Notation des Informatik-Duden) sei gegeben:

```
1  procedure quicksort(links, rechts : integer)
2  var i, j, x : integer;
3  begin
4      i := links;
5      j := rechts;
6      if j > i then begin
7          x := A[links];
8          repeat
9              while A[i] < x do i := i+1;
10             while A[j] > x do j := j-1;
11             if i < j then begin
12                 tmp := A[i]; A[i] := A[j]; A[j] := tmp;
13                 i := i+1; j := j-1;
14             end
15             until i > j;
16             quicksort(links, j);
17             quicksort(i, rechts);
18         end
19     end
```

Umsetzung in Java:

```
6  public static void quicksort(int[] A, int links, int rechts) {
7      int i = links;
8      int j = rechts;
9      if (j > i) {
10         int x = A[links];
11         do {
12             while (A[i] < x) {
13                 i = i + 1;
14             }
15             while (A[j] > x) {
16                 j = j - 1;
17             }
18             if (i <= j) {
19                 int tmp = A[i];
20                 A[i] = A[j];
21                 A[j] = tmp;
22                 i = i + 1;
23                 j = j - 1;
24             }
25             // Java verfügt über keine do-until Schleife.
26             // Wir verwenden eine do-while-Schleife mit einem umgedrehten Test
27             // until i > j -> while (i <= j)
28         } while (i <= j);
29         quicksort(A, links, j);
30         quicksort(A, i, rechts);
31     }
32 }
```

Der initiale Aufruf der Prozedur lautet:

```
1  quicksort(0,n-1)
```

- (a) Sortieren Sie das folgende Feld der Länge 7 mittels des Algorithmus. Notieren Sie jeweils alle Aufrufe der Prozedur quicksort mit den konkreten Parameterwerten. Geben Sie zudem für jeden Aufruf der Prozedur den Wert des in Zeile 7 gewählten Elements an.

27 13 21 3 6 17 44 42

```

1  quicksort(0, 6)
2  27 32 3 6 17 44 42
3  x: 27
4  quicksort(0, 2)
5  17 6 3 32 27 44 42
6  x: 17
7  quicksort(0, 1)
8  3 6 17 32 27 44 42
9  x: 3
10 quicksort(0, -1)
11 3 6 17 32 27 44 42
12 quicksort(1, 1)
13 3 6 17 32 27 44 42
14 quicksort(2, 2)
15 3 6 17 32 27 44 42
16 quicksort(3, 6)
17 3 6 17 32 27 44 42
18 x: 32
19 quicksort(3, 3)
20 3 6 17 27 32 44 42
21 quicksort(4, 6)
22 3 6 17 27 32 44 42
23 x: 32
24 quicksort(4, 3)
25 3 6 17 27 32 44 42
26 quicksort(5, 6)
27 3 6 17 27 32 44 42
28 x: 44
29 quicksort(5, 5)
30 3 6 17 27 32 42 44
31 quicksort(6, 6)
32 3 6 17 27 32 42 44
33 3 6 17 27 32 42 44

```

- (b) Angenommen, die Bedingung $j > i$ in Zeile 6 des Algorithmus wird ersetzt durch die Bedingung $j \geq i$. Ist der Algorithmus weiterhin korrekt? Begründen Sie Ihre Antwort.

```

1  quicksort(0, 6)
2  27 32 3 6 17 44 42
3  x: 27
4  quicksort(0, 2)
5  17 6 3 32 27 44 42
6  x: 17
7  quicksort(0, 1)
8  3 6 17 32 27 44 42
9  x: 3
10 quicksort(0, -1)
11 3 6 17 32 27 44 42
12 quicksort(1, 1)
13 3 6 17 32 27 44 42

```

```

14  x: 6
15  quicksort(1, 0)
16  3 6 17 32 27 44 42
17  quicksort(2, 1)
18  3 6 17 32 27 44 42
19  quicksort(2, 2)
20  3 6 17 32 27 44 42
21  x: 17
22  quicksort(2, 1)
23  3 6 17 32 27 44 42
24  quicksort(3, 2)
25  3 6 17 32 27 44 42
26  quicksort(3, 6)
27  3 6 17 32 27 44 42
28  x: 32
29  quicksort(3, 3)
30  3 6 17 27 32 44 42
31  x: 27
32  quicksort(3, 2)
33  3 6 17 27 32 44 42
34  quicksort(4, 3)
35  3 6 17 27 32 44 42
36  quicksort(4, 6)
37  3 6 17 27 32 44 42
38  x: 32
39  quicksort(4, 3)
40  3 6 17 27 32 44 42
41  quicksort(5, 6)
42  3 6 17 27 32 44 42
43  x: 44
44  quicksort(5, 5)
45  3 6 17 27 32 42 44
46  x: 42
47  quicksort(5, 4)
48  3 6 17 27 32 42 44
49  quicksort(6, 5)
50  3 6 17 27 32 42 44
51  quicksort(6, 6)
52  3 6 17 27 32 42 44
53  x: 44
54  quicksort(6, 5)
55  3 6 17 27 32 42 44
56  quicksort(7, 6)
57  3 6 17 27 32 42 44
58  3 6 17 27 32 42 44

```

- (c) Angenommen, die Bedingung $i \leq j$ in Zeile 11 des Algorithmus wird ersetzt durch die Bedingung $i < j$. Ist der Algorithmus weiterhin korrekt? Begründen Sie Ihre Antwort.

bleibt hängen

```

1  quicksort(0, 6)
2  27 32 3 6 17 44 42
3  x: 27
4  quicksort(0, 2)
5  17 6 3 32 27 44 42
6  x: 17
7  quicksort(0, 1)
8  3 6 17 32 27 44 42

```

9 x: 3

- (d) Wie muss das Feld A gestaltet sein, damit der Algorithmus mit der geringsten Anzahl von Schritten terminiert? Betrachten Sie dazu vor allem Zeile 7. Begründen Sie Ihre Antwort und geben Sie ein Beispiel.
- (e) Die rekursiven Aufrufe in den Zeilen 16 und 17 des Algorithmus werden zur Laufzeit des Computers auf dem Stack verwaltet. Die Anzahl der Aufrufe von quicksort auf dem Stack abhängig von der Eingabegröße n sei mit $s(n)$ bezeichnet. Geben Sie die Komplexitätsklasse von $s(n)$ für den schlimmsten möglichen Fall an. Begründen Sie Ihre Antwort.