

AUFGABE 12:

Der Median einer sortierten Folge $x_1 \leq x_2 \leq \dots \leq x_n$ ist das Element $x_{\lceil n/2 \rceil}$. Geben Sie für die folgenden Probleme schleifenfreie Algorithmen an, die als Operationen nur *Vertauschen* und *Vergleichen* benutzen. Mit wie vielen solcher Operationen kommen Sie aus? Begründen Sie die Korrektheit Ihrer Algorithmen.

- Sortieren Sie fünf Zahlen a,b,c,d,e.
- Bestimmen Sie den Median von sieben Zahlen und benutzen Sie a) dazu.

Bemerkung: Ein solcher schleifenfreier Algorithmus bekommt die Eingabe x_1, x_2, \dots, x_n in einem Array $A[1 : n]$. Er besteht aus einer Folge von Operationen der Form:

if $A[i] < A[j]$ then *vertausche*($A[i], A[j]$) $A[i] \longrightarrow A[j]$


Beispiel für das Sortieren von drei Zahlen:

if $A[2] < A[1]$ then $vertausche(A[1], A[2])$ $A[1] \longrightarrow A[2]$

```

if A[3] < A[2] then vertausche(A[2], A[3])

```



```

graph LR
    A1[A[1]] --> A3[A[3]]
    A2[A[2]] --> A3

```

if $A[2] < A[1]$ then $vertausche(A[1], A[2])$ $A[1] \longrightarrow A[2] \longrightarrow A[3]$

Um die Korrektheit zu beweisen, kann man zum Beispiel Pfeildiagramme benutzen. Dabei steht $A[i] \rightarrow A[j]$ für $A[i] \leq A[j]$, wie im obigen Beispiel gezeigt.

Lösung: Die Aufgabe läßt verschiedene Interpretationen zu. Zum einen könnte man verbieten, daß Verzweigungen vorkommen, d.h. die zu benutzenden Befehle haben alle die Form `if $A[i] < A[j]$ then vertausche($A[i]$, $A[j]$)`. Somit werden Blockungen ausgeschlossen und die gesuchte Lösung ist ein Sortiernetzwerk, daß unabhängig von den betrachteten Werten ein Feld sortiert (man nennt diese Art von Sortierverfahren auch 'oblivious' [vergesslich], weil sie die zu sortierenden Schlüssel sofort wieder vergessen). Sehr bekannte Vertreter dieser Netzwerke sind Batchers-Sortiernetzwerk, Beneß-Netzwerk oder das Butterfly-Netzwerk. In der gewählten Darstellungsform stellen alle vertikalen Linien die zu sortierenden Zahlen dar und alle Knotenpunkte die Austauschoperationen. Abbildung 1 zeigt ein Butterfly-Netzwerk.

Die andere Möglichkeit der Lösung der Aufgabe würde das Blocken von Befehlen erlauben. Somit ist es möglich, entsprechend einer Vergleichsoperation zu verzeigen. Somit können Operationen eingespart werden. Folgender Algorithmus sortiert 5 Zahlen mit 7 Vergleichen. Der Einfachheit halber werden die Zahlen mit a , b , c , d und e bezeichnet.

$\text{SORT}(a, b, c, d, e)$

Description: Diese Prozedur sortiert die 5 Zahlen der Eingabe mit 7 Vergleichen.

```
(1)  if  $a > b$ 
(2)    TAUSCHE( $a, b$ )
(3)  if  $c > d$ 
(4)    TAUSCHE( $c, d$ )
(5)  if  $b > d$ 
(6)    TAUSCHE( $a, c$ )
(7)    TAUSCHE( $b, d$ )
(8)  if  $e < b$ 
(9)    if  $e < a$ 
(10)     TAUSCHE( $a, e$ )
(11)     TAUSCHE( $e, b$ )
(12)     TAUSCHE( $e, d$ )
(13) else
(14)  if  $e < d$ 
(15)     TAUSCHE( $e, d$ )
(16)  if  $c < b$ 
(17)    if  $c < a$ 
(18)     TAUSCHE( $a, c$ )
(19)     TAUSCHE( $b, c$ )
(20) else
(21)  if  $c > d$ 
(22)     TAUSCHE( $c, d$ )
```

Die Korrektheit des Algorithmus wird aus den untenstehenden Diagrammen in Abbildung 2 ersichtlich.

Auch für Aufgabe b sind mehrere Interpretationen möglich. Die einfachste (aber nicht die optimale bezüglich der Anzahl der Vergleiche) ist die Sortierung der 7 Zahlen mit Hilfe der oben entwickelten Prozedur SORT (siehe SORTSIEBEN).

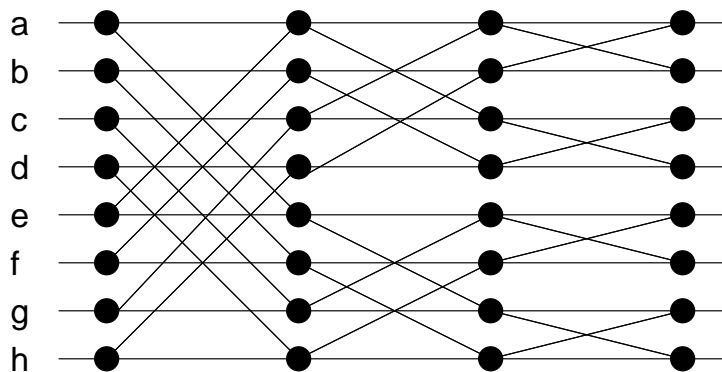


Abbildung 1: Die Abbildung zeigt eine Butterfly-Netzwerk. In jedem Level werden $n/2$ Zahlen miteinander verglichen und wenn nötig vertauscht.

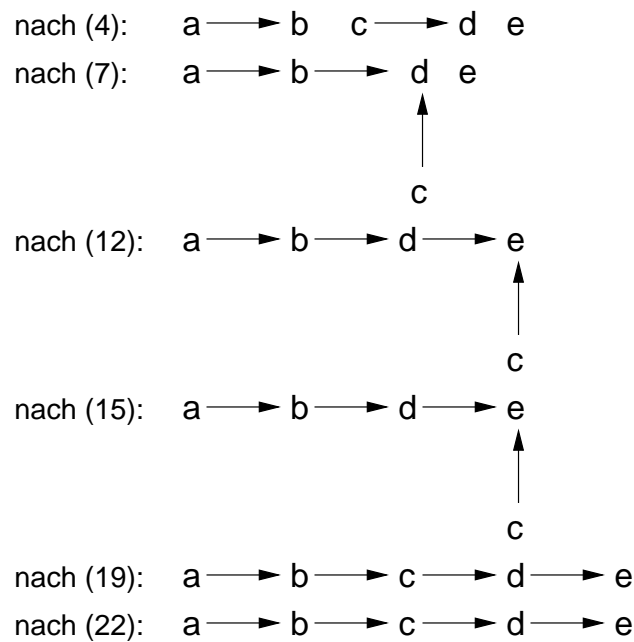


Abbildung 2: Die Abbildung zeigt die Korrektheit des Algorithmus SORT anhand der Relationen der einzelnen Elementen zueinander nach jedem vertauschen von Elementen.

$\text{SORTSIEBEN}(a, b, c, d, e, f, g)$

Description: Dieser Algorithmus bestimmt den Median von sieben Zahlen, indem die eingabe über eine Prozedur zum Sortieren von 5 Zahlen verwendet wird. Die Elemente der Eingabe sind hinterher sortiert (siehe STOOGESORT).

- (1) $\text{SORT}(a, b, c, d, e)$
- (2) $\text{SORT}(c, d, e, f, g)$
- (3) $\text{SORT}(a, b, c, d, e)$
- (4) **return** d

Die zweite Möglichkeit erlaubt wieder das Einsortieren und Blocken von Anweisungen. Der folgende Algorithmus benötigt insgesamt 11 Vergleiche zum Finden des Medians.

MEDIANSIEBEN(a, b, c, d, e, f, g)

Description: Diese Prozedur bestimmt den Median von 7 Elementen. Das geschieht durch Einsortieren der letzten beiden Elemente in ein vorher sortiertes Feld von 5 Elementen (1). Insgesamt werden maximal 4 Vergleiche (ohne SORT) benötigt.

```
(1) SORT( $a, b, c, d, e$ )
(2) if  $f < c$ 
(3)   if  $g < c$ 
(4)     return MAX( $b, f, g$ )
(5)   else
(6)     return  $c$ 
(7) else
(8)   if  $g < c$ 
(9)     return  $c$ 
(10) else
(11)  return MIN( $g, f, d$ )
```