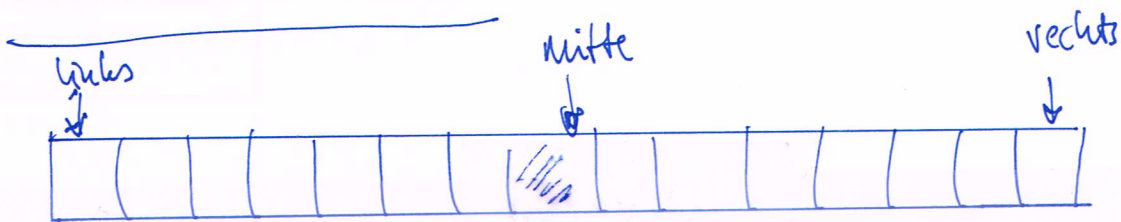


Max-Teilfeld-Problem

Algo + DS
10.06.21



FIND-MAXIMUM-SUBARRAY(A , $links$, $rechts$)

```
1 if  $rechts == links$ 
2   return ( $links$ ,  $rechts$ ,  $A[links]$ ) // Basisfall: nur ein Element
3 else  $mitte = \lfloor (links + rechts) / 2 \rfloor$ 
4 ① ( $links-links$ ,  $links-rechts$ ,  $linke-summe$ ) =
   FIND-MAXIMUM-SUBARRAY( $A$ ,  $links$ ,  $mitte$ )
5 ② ( $rechts-links$ ,  $rechts-rechts$ ,  $rechte-summe$ ) =
   FIND-MAXIMUM-SUBARRAY( $A$ ,  $mitte + 1$ ,  $rechts$ )
6 ( $mittig-links$ ,  $mittig-rechts$ ,  $mittige-summe$ ) =
   FIND-MAX-CROSSING-SUBARRAY( $A$ ,  $links$ ,  $mitte$ ,  $rechts$ )
7 if  $linke-summe \geq rechte-summe$  und  $linke-summe \geq mittige-summe$ 
8   return ( $links-links$ ,  $links-rechts$ ,  $linke-summe$ )
9 elseif  $rechte-summe \geq linke-summe$  und  $rechte-summe \geq mittige-summe$ 
10  return ( $rechts-links$ ,  $rechts-rechts$ ,  $rechte-summe$ )
11 else return ( $mittig-links$ ,  $mittig-rechts$ ,  $mittige-summe$ )
```

FIND-MAX-CROSSING-SUBARRAY(A , $links$, $mitte$, $rechts$)

```
1  $linke-summe = -\infty$ 
2  $summe = 0$ 
3 for  $i = mitte$  downto  $links$ 
4    $summe = summe + A[i]$ 
5   if  $summe > linke-summe$ 
6      $linke-summe = summe$ 
7      $max-links = i$ 
8  $rechte-summe = -\infty$ 
9  $summe = 0$ 
10 for  $j = mitte + 1$  to  $rechts$ 
11    $summe = summe + A[j]$ 
12   if  $summe > rechte-summe$ 
13      $rechte-summe = summe$ 
14      $max-rechts = j$ 
15 return ( $max-links$ ,  $max-rechts$ ,  $linke-summe + rechte-summe$ )
```

