

$$1.) \quad T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + \Theta(1)$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + \Theta(1)$$

$$T(n) = 2^k T\left(\frac{n}{2^k}\right) + k \Theta(1)$$

$$\frac{n}{2^k} \stackrel{!}{=} 1 \Rightarrow k = \text{ld}(n)$$

$$T(n) = n \cdot \Theta(1) + \text{ld}(n) + \Theta(1)$$

$$= \Theta(n) + \Theta(\text{ld}(n)) = \Theta(n)$$

2.) IB: $n=1$ Algo(A, 1, 1) \rightarrow A(1) = größte Zahl
 $n=2$ Algo(A, 1, 2) \rightarrow A(2) = größte Zahl

IA: Algo liefert für n Zahlen in A die größte Zahl

IS: Algo(A, 1, n+1)



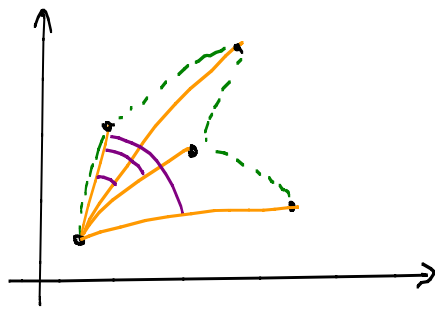
Algo(A, 1, p)

\downarrow
 x_1 größte Zahl

\downarrow
 x_2 größte Zahl

nehmen $\max(x_1, x_2) \rightarrow$ größte Zahl

3.)



1.) Punkt mit einem suchen $O(n)$

2.) berechne $\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$ $O(n)$

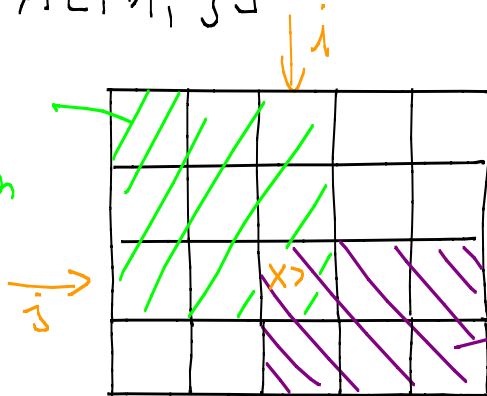
3.) Vergleich $O(n \log n)$

4.) Punkte verbinden

5.) $A[i, j] \leq A[i, j+1]$

$A[i, j] \leq A[i+1, j]$

Hier kann
x nicht mehr
sein, wenn
 $A(i, j) \leq x$



Hier kann x nicht sein
wenn $A(i, j) \geq x$

```

i=1; j=12
while(A[i,j] != x & i <= n & j >= 1)
  while(A[i,j] > x & j >= 1)
    j = j-1;
  while(A[i,j] < x & i <= n)
    i = i+1;
end;
if(i > n) or (j < 1) --> kein x vorhanden
ansonsten ja
  
```

Laufzeit: $2n \Rightarrow O(n)$