Today → Selection in $O(n)$

```
QuickSort(A):
 if len(A) ≤ 1:                    } aiming for O(n)
    return A
 else: Find the median
    Use Partition to move all small things to front } Θ(n)
    and big things to back
    Recurse on smalls      T(  )
    Recurse on bigs        T(  )
```

if we pick the median as pivot: $T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$

else worst case either biggest or smallest num: $T(n) = T(n-1) + \Theta(n) = \Theta(n^2)$

Selection$(A[1,\dots,n], i)$ → Find the i^{th} smallest item in A $(1 \le i \le n)$

$O(n)$ 1. Find the median of each group of 5 → new Pivot candidate now we have $T(n/5)$

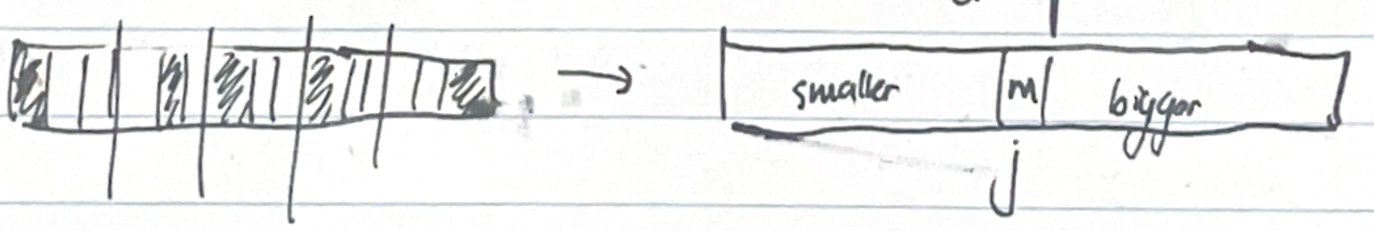
$T(n/3)$ 2. Make a list out of them, recursively find the median of that list m

$O(n)$ 3. Use Partition to split the list into things smaller than the median m and bigger than m. Let j be the index of the start of the bigger things.

4. If $i = j$: return m

3 cases else if $i < j$: return Selection$(A[1:j], i)$ $T(\frac{2n}{3})$

else: return Selection$(A[j:n], i-j)$



* 2 things per median smaller than m that are smaller than m

$\Rightarrow n/3$ things smaller than m

↳ lower bound on deletions.

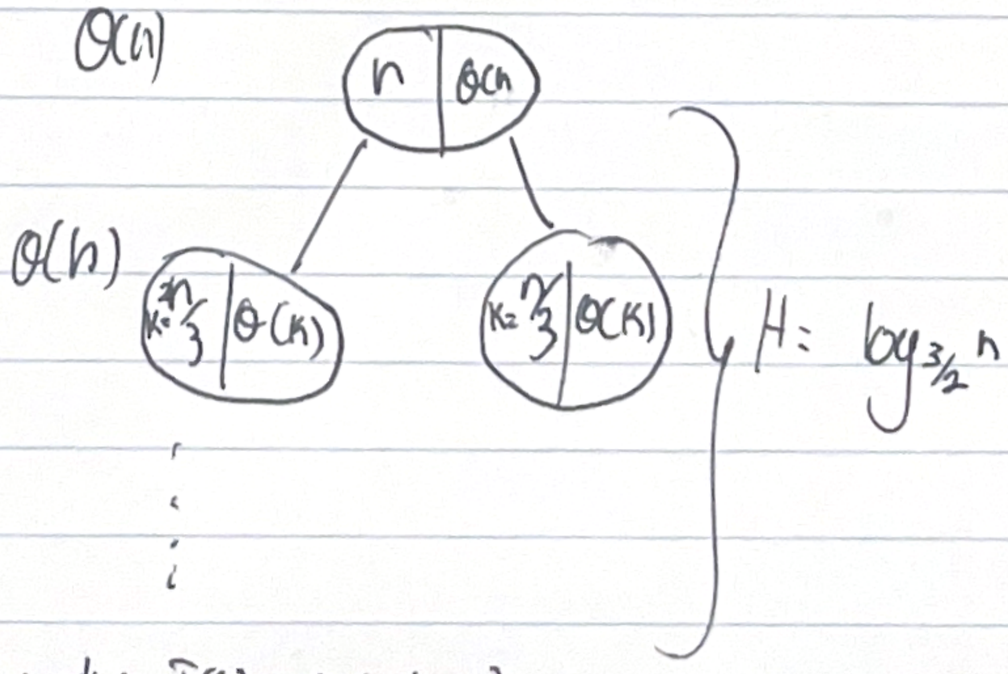
$$T(n) = T(\tfrac{2n}{3}) + T(\tfrac{n}{3}) + \Theta(n)$$

with $n=5$ $T(n) = T(\frac{3n}{5}) + T(\frac{n}{5}) + \Theta(n)$

$T(\frac{7n}{10})$

* Check ↳ why this is true

$O(n)$



$O(n)$

Total work: $T(n) = \Theta(n \log n)$