

SELECT (list, i) {

Note Title

10/8/2007

// i = position of desired value.

① Compute MoM_3 .

↳ by calling **SELECT** (list of medians, $\frac{n}{6}$)

example: MoM_3 of $\{2, 6, 5, 12, 11, 10\}$
is **SELECT** ($\{5, 11\}$, 1) = 5

② Partition around MoM_3 and
let pos_{MoM_3} = position of MoM_3 after
partition.

③ If $i > pos_{MoM_3}$,
call **SELECT** (" $> MoM_3$ " list, $i - pos_{MoM_3}$)

If $i < pos_{MoM_3}$,
call **SELECT** (" $< MoM_3$ " list, i)

If $i == pos_{MoM_3}$,
return MoM_3 .

Using sublists of size 3
gives us $O(n \log n)$ runtime.

→ No better than sorting!

But often divide and conquer
algorithms do save us time,

if we eliminate enough of the
input in each recursive step

Can we eliminate MORE
values before we recurse??

How about if we break the
input into sublists of size 5?

// NEW VERSION: SUBLISTS OF SIZE 5

SELECT (list, i) {

// i = position of desired value.

10/8/2007

① Compute MoM_5 .

↳ by calling **SELECT** (list of medians, $\frac{n}{10}$)

example: MoM_5 of {4, 1, 3, 7, 2, 9, 8, 5, 11, 6}
is **SELECT** ({3, 8}, 1) = 3

② Partition around MoM_5 and
let pos_{MoM_5} = position of MoM_5 after
partition.

③ If $i > pos_{MoM_5}$,
call **SELECT** ("> MoM_5 " list, $i - pos_{MoM_5}$)

If $i < pos_{MoM_5}$,
call **SELECT** ("< MoM_5 " list, i)

If $i == pos_{MoM_5}$,
return MoM_5 .

What is the RT of ① and ②?

① Compute Mom_5

Find median
of 5 elements.

- Time to find medians
of each sublist =

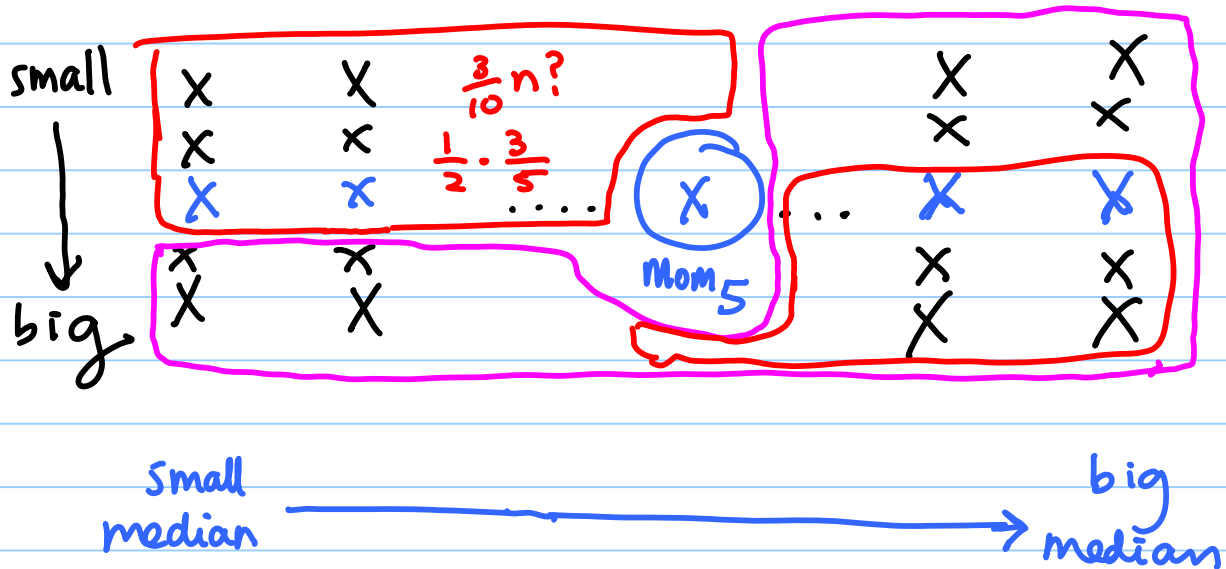
~~9~~ 10 comparisons $\frac{n}{5}$ times = $O(n)$

- Time to compute
 Mom_5 from list of medians
of size $\frac{n}{5}$ done recursively:
 $T(\frac{n}{5})$

② Time to partition: $O(n)$

But how hard is it to
compute step ③ — the
main recursive call?

What is the run time of ③?



Medians in blue.

Dead center is the median of medians.

Definitely smaller than Mom_5 :

$$\begin{aligned} & \left(3 \text{ values in } \frac{1}{2} \text{ lists} \right) - 1 \leftarrow Mom_5 \\ & = 3 \cdot \frac{1}{2} \cdot \frac{n}{5} = \frac{3n}{10} - 1 \end{aligned}$$

Worst-case size of " $> Mom_5$ ":

$$n - \left(\frac{3n}{10} - 1 \right) - 1 \leftarrow Mom_5 = \frac{7n}{10}$$

$$T(n) = \underbrace{\text{time to compute MoM}_3}_{\text{time to find medians + of each sublist}} + \underbrace{\text{time to partition}}_{\text{time to compute MoM}_3 \text{ from list of medians}} + T\left(\frac{7n}{10}\right)$$

time to
find medians +
of each sublist

↓

$O(n)$

$$\frac{10}{10} \cdot \frac{n}{5} = 2n$$

time to
compute
MoM₃

from list
of medians

list size: $\frac{n}{5}$

done recursively

$T\left(\frac{n}{5}\right)$

↓
 $O(n)$

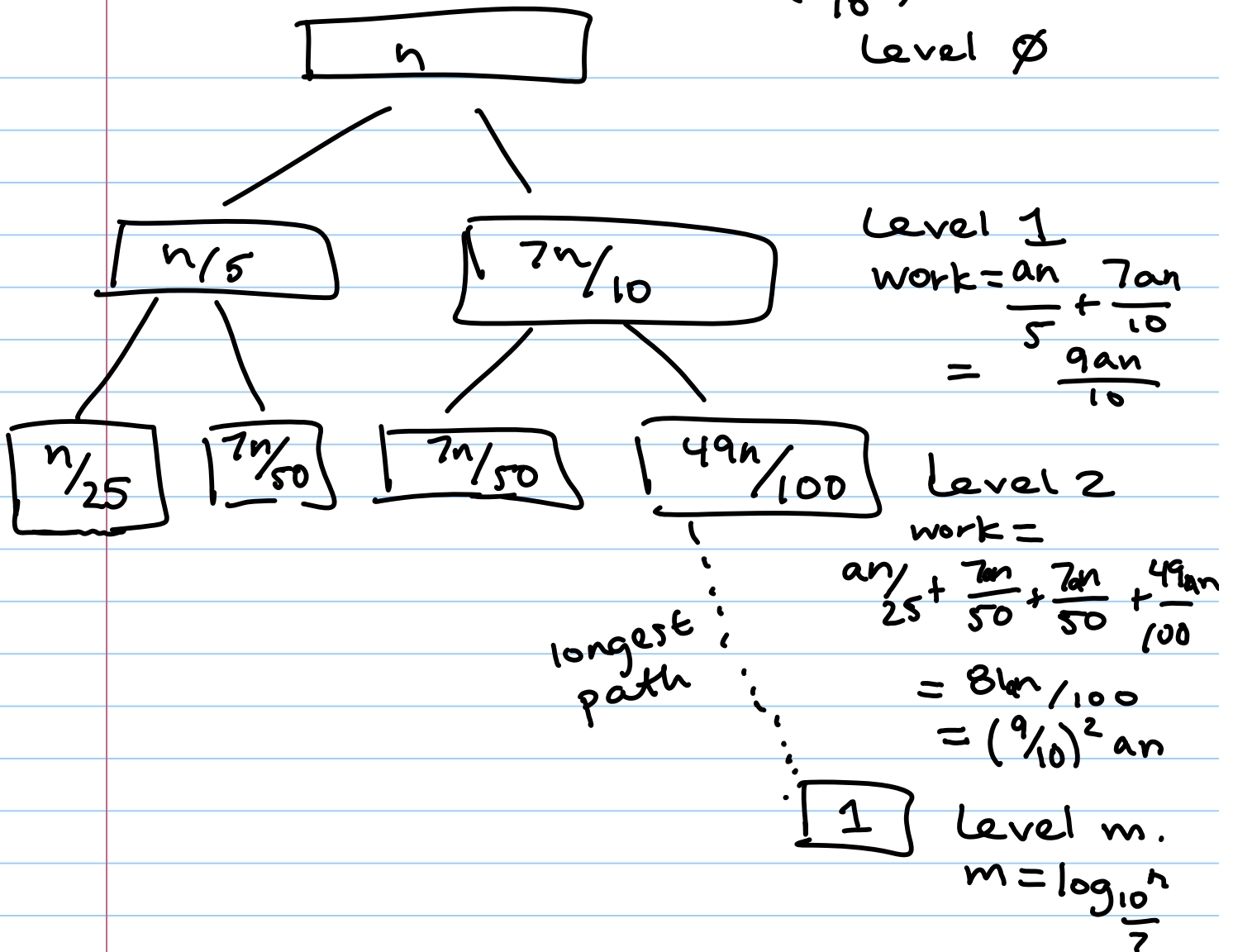
$n-1 \approx n$

$$T(n) = O(n) + T\left(\frac{n}{5}\right) + O(n) + T\left(\frac{7n}{10}\right)$$

$$T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + O(n)$$

Let's solve this...

$$T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + an$$



$$T(n) \leq \sum_{i=0}^{\log_{10 \frac{7}{5}} n} a \left(\frac{9}{10}\right)^i n$$

$$T(n) \leq \sum_{i=0}^{\log_{10/7} n} a \left(\frac{9}{10} \right)^i n$$

$$= a n \left(\sum_{i=0}^{\log_{10/7} n} \left(\frac{9}{10} \right)^i \right)$$

$$\leq a n \left(\frac{1}{1 - \frac{9}{10}} \right) = 10 a n .$$

$$T(n) \in O(n)$$

What about big Ω

$$T(n) \geq \sum_{i=0}^{\log_5 n} \left(\frac{9}{10}\right)^i a_n$$

$$= \left(\frac{\left(\frac{9}{10}\right)^{\log_5 n + 1} - 1}{\frac{9}{10} - 1} \right) a_n$$

$$= \left(\frac{\left(\frac{9}{10}\right) \left(\frac{9}{10}\right)^{\log_5 n} - 1}{-\frac{1}{10}} \right) a_n$$

$$= -10 \left[\left(\frac{9}{10}\right) \left(n^{\log_5 \frac{9}{10}}\right) - 1 \right] a_n$$

$$= -9 n^{\log_5 \frac{9}{10}} a_n + 10 a_n$$

$$-9an \left(n^{\log_5 9/10} \right) + 10an$$

$$= -9an^{\log_5 9/10 + 1} + 10an$$

$\log_5 9/10$ is negative

$$\log_5 9/10 + 1$$

$$= \log_5 9/10 + \log_5 5$$

$$= \log_5 (9/10)(5) \quad \begin{matrix} 5^y = 4\frac{1}{2} \\ y < 0 \end{matrix}$$

$$= \log_5 9/2 \approx 0.93$$

$$T(n) \geq 10an - 9an^{0.93}$$

$$T(n) \in \Omega(n)$$

$$T(n) \in \Theta(n)$$

means $T(n) = cn$.

Can we figure out the value
of c ?

Prove $T(n) = cn$

Inductive Hypothesis

$$\forall i \ 10 \leq i < n \quad T(i) = ci$$

Inductive Step.

$$T(n) = cn$$

$$T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + an = cn.$$

apply
IH.

$$\frac{cn}{5} + \frac{7cn}{10} + an = cn$$

$$\frac{cn}{5} + \frac{cn}{10} + an = cn$$

value for a ?

an = time to find medians + time to partition

$$an \leq 2n + n$$

$$a \leq 3$$

$$\text{set } a = 3$$

$$\frac{cn}{5} + \frac{7cn}{10} + 3n = cn$$

$$2cn + 7cn + 30n = 10cn$$

$$-cn = -30n$$

$$c = 30$$

So our algorithm has runtime $30n$.

note:

algorithms better than
 $3n$ exist!

but:

it has been proven
that no algorithm can do
better than $2n$.

Actual Lower Bound

↳ Unknown!

