Selection problem:

Given a set $S = \{x_1, x_2 \cdots x_n\}$ of n distinct numbers and $k \in [n]$
Find the k'th element

k'th element : $x_{i_1} < x_{i_2} < \cdots < x_{i_{k-1}} < x_{i_k} < \cdots\cdots < x_{i_n}$

strategy: pick one element x_j called the pivot and arrange around x_j

$$\overset{A}{\boxed{\quad < \quad}}\ x_j\ \overset{B}{\boxed{\quad < \qquad}}$$

If we can pick x_j s.t. $\min\{|A|, |B|\} \geq c \cdot n$
then we can obtain a linear algorithm.

when $k = \frac{n+1}{2}$ (n odd) we call
the k'th element the median

$n = 2r+1$

$r+1$

a lower bound for finding the median

use acyclic orientations:

If m <u>is</u> median (we have found it)

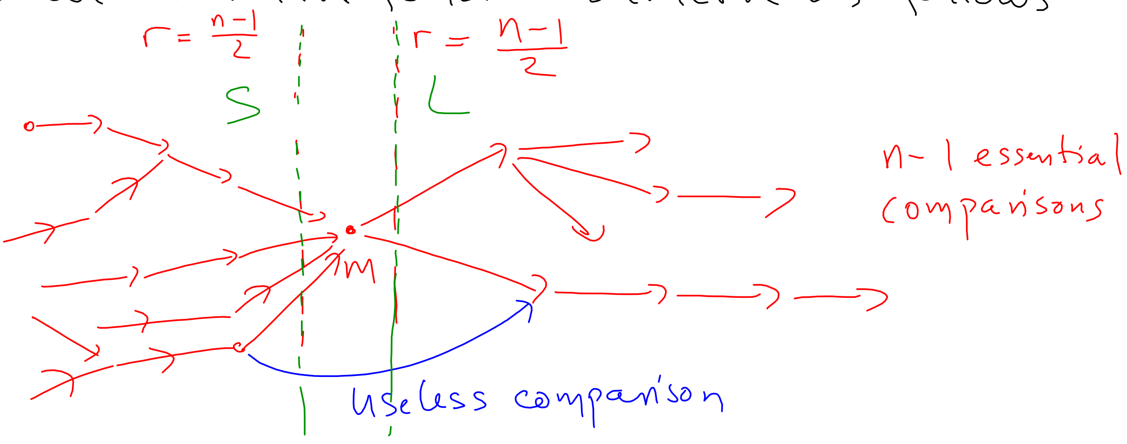
then every algorithm which determines m as the median must have discovered r elements s.t. these are smaller than m

and r - - - - - - - larger - - -

if $x < m$ then $x \to \circ \rightsquigarrow \circ \to m$

and if $y > m$ $m \rightsquigarrow \circ \to y$

So we must have found a structure as follows



adversary wants to force $\mathcal{A}$ to make many useless comparisons.

We maintain sets

U := never compared (initially $|U| = n$)

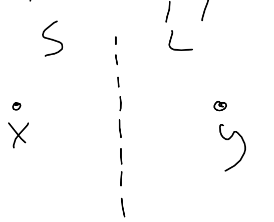
L := larger than median

S := smaller ....

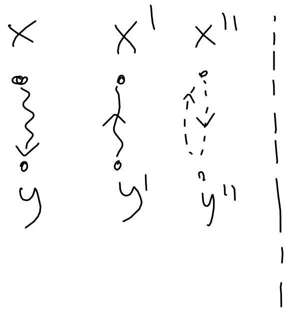
m = median (will always be last element to leave U)

$\mathcal{A}$: Compare(x, y) :

$x, y \in U$:	$x < y$, $x \to S$, $y \to L$	$x<y$
$x \in U$, $y \in S$	$x \to L$	$y < x$
$x \in U$, $y \in L$	$x \to S$	$x < y$



$x < y$



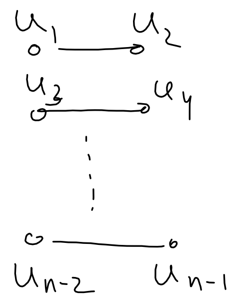
$x < y$

$x' > y'$

$x'' < y''$

adversary can force at least $\frac{n-1}{2}$ useless comparisons

(worst case for adversary:



$\left\{ \begin{array}{l} A \text{ needs at least } n-1 \\ \text{essential comparisons and we} \\ \text{force at least } \frac{n-1}{2} \text{ useless} \end{array} \right.$

$\Downarrow$

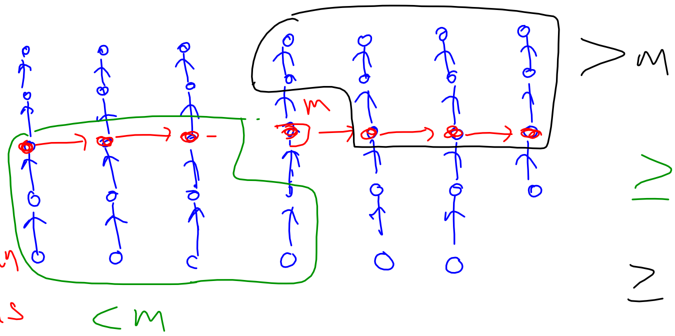
A uses at least $n-1 + \frac{n-1}{2} = \frac{3n}{2} - \frac{3}{2}$ comp.

Finding the k'th element in linear time.

Idea use medians the acheive a good partion around pivot

partion S into $\lceil \frac{n}{5} \rceil$ sets ($S_1, S_2, \dots, S_{\lceil \frac{n}{5} \rceil}$) with 5 elements in all except possible one

1. Sort each set S_i
2. isolate medians
3. Find median of medians $\longrightarrow m$



$\geq \frac{3n}{10} - 6$ element

$\geq \frac{3n}{10} - 6$ elements

$T(n)$ = # comparisons :

$$T(n) = \begin{cases} \Theta(n) & n \leq 140 \\ T(\lceil \frac{n}{5} \rceil) + T(\frac{7n}{10} + 6) + O(n) & n > 140 \end{cases}$$

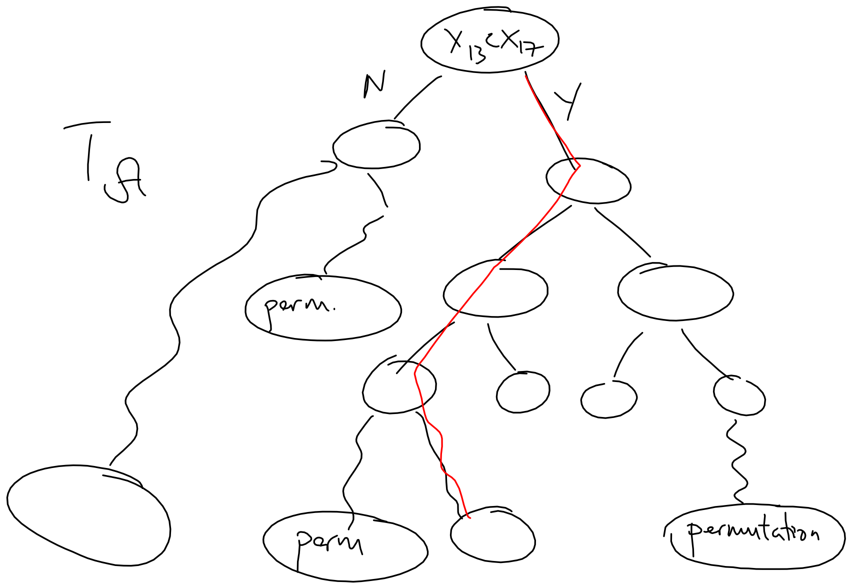
$\Longrightarrow$ $T(n) \leq cn$ for c suitably large

Lower bound for comparison-based sorting.

Known (e.g. from DM507/Cormen book):
Mergesort sorts using $\Theta(n\log n)$ comparisons

$\mathcal{A}$ sorts by comparisons
$\Downarrow$ represent $\mathcal{A}$ as a binary decission tree

$T_{\mathcal{A}}$



$\mathcal{A}$ must be able to determine any permutation as output (a leaf)

$\Rightarrow \geq n!$ leaves in $T_{\mathcal{A}}$

$T_{\mathcal{A}}$ is binary so at most 2^h leaves h height

so $2^h \geq n! \Rightarrow h \geq \log_2 n! \sim n\log n - cn$

Adversary strategy 1:
(powerfull adversary)

always maintain a list $\mathcal{L}$ of all permutations consistent with answers so far.

When answering $x<y$? in step i

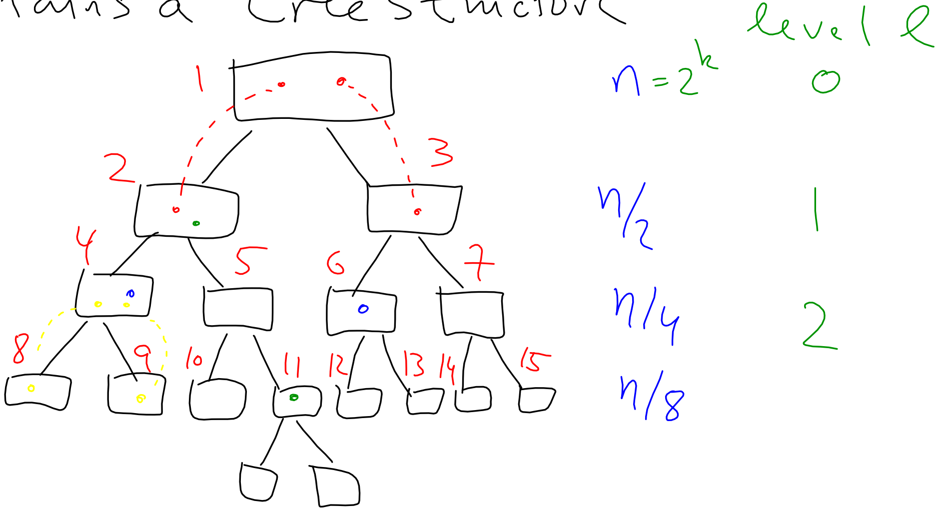
answer s.t. $\mathcal{L}^i \geq \mathcal{L}^{i-1}/2$

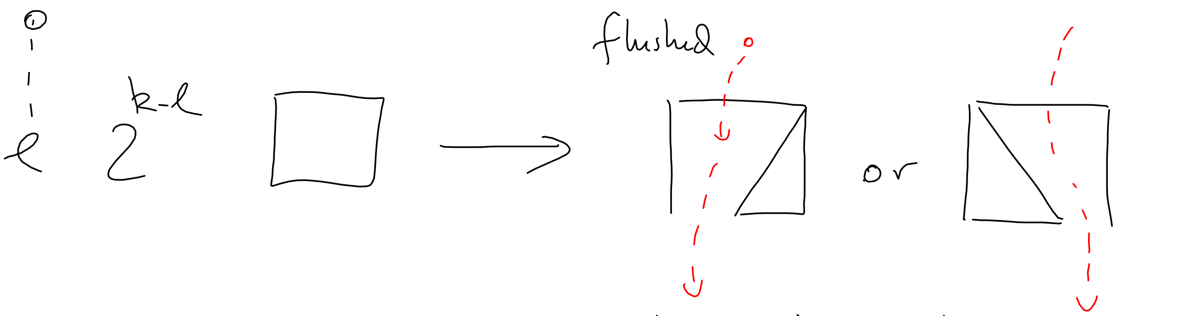
$\Downarrow$ at least $\log_2 n!$ questions forced

$\longrightarrow \Omega(n \log n)$

Adversary 2 (much less powerfull)

maintains a tree structure





if left or right subtree already has

$$2^{k-l-1} = \frac{2^{k-l}}{2} \text{ elements}$$

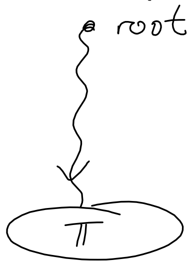
adversary only moves an element x to a lower box if 1. x is in a comparison

or 2 the box that x enters is "flushed"

$\Rightarrow$ adversary can force $\frac{n}{2}\log_2 n$ comparisons.

average no of comparisons.

comparisons for fixed perm. π:



epl = sum of all lengths of paths from root to leaf

epl minimum when T is almost balanced