Maximum among n elements

compare first element x with second y
and keep $z = \max\{x, y\}$
compare z with next and keep max etc.

Finds max in $\boxed{n-1 \text{ comparisons}}$

complexity is in # comparisons

Best possible: if x has not been compared with anything, then it can still be max

Finding max and min: S set of distinct n numbers

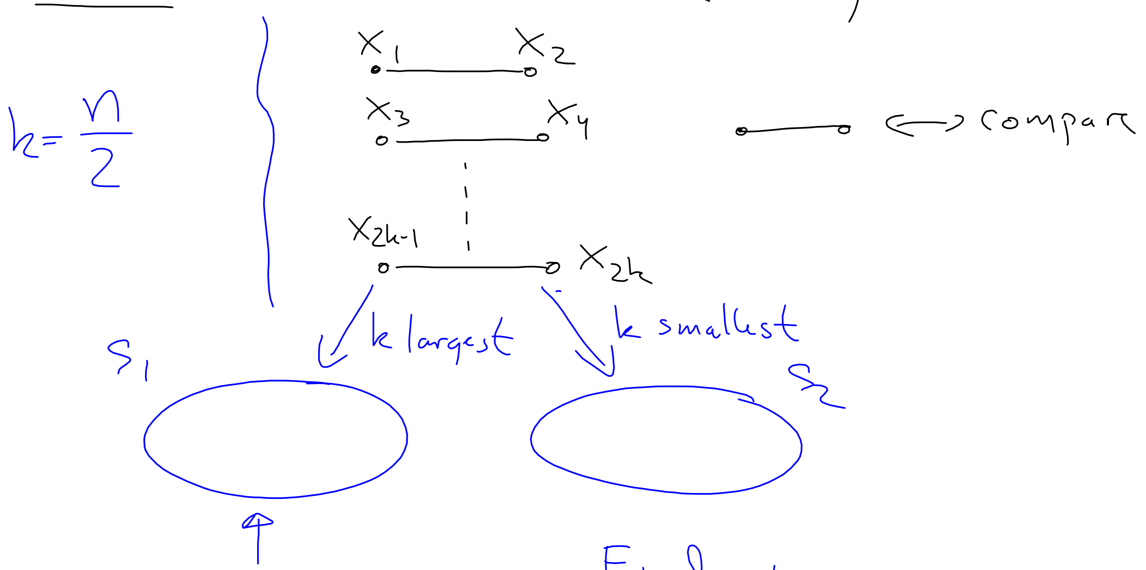
Naive alg:

- Find max in S call it z $\quad n-1$ comparisons
- Find min in $S-z$ $\quad n-2$

$$\frac{\qquad}{2n-3}$$

Different strategy:

Case 1 $n = 2k$ (even)



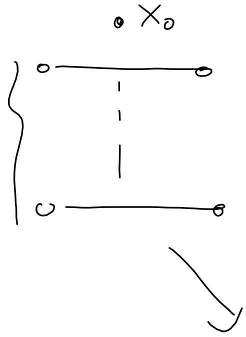
Find max $k-1 = \frac{n}{2}-1$

Find min $k-1 = \frac{n}{2}-1$

Total # comparisons $= 2\left(\frac{n}{2}-1\right)+\frac{n}{2} = \frac{3}{2}n-2$

<u>Case 2</u> $n = 2k+1$

$k = \frac{n-1}{2}$



S_1 $\qquad\qquad$ S_2

$k-1 = \frac{n-3}{2}$ $\qquad\qquad$ $k-1 = \frac{n-3}{2}$

$$\frac{n-1}{2} + n - 3 = \frac{3n}{2} - \frac{7}{2}$$

Compare x_0 with max and possibly also min

+2 comparisons

total $\frac{3n}{2} - \frac{7}{2} + 2 = \frac{3n}{2} - \frac{3}{2}$

Adversary strategy

Idea: Every algorithm $\mathcal{A}$ for max + min

x = max returned
y = min returned

$\Longrightarrow$ x must have been larger than all other
so $\geq n-1$ pieces of information
do $\geq n-1$ info before y is known.
$\longrightarrow$ $\geq 2n-2$ pieces of info must be collected.

how many times can $\mathcal{A}$ obtain 2 pieces of info by one comparison?

at most $\left\lfloor \frac{n}{2} \right\rfloor$: only if x and y can still be both max and min

stategy for adversary

when comparing x and y:

notation

a → b : a < b

z : z never compared with anything

z : z has lost all comp

z : z has won all comp

z : z has both won and lost

---

assume A compares a and b

pieces of info to A		answer
2	1.	a < b
1	2.	b < a
1	3.	b > a
1	4.	b > a
1	5.	a < b
1	6.	a < b
1		a < b
0		a < b !

so that consistency is maintained.

Conclusion $\mathcal{A}$ can only collect 2 pieces of info in case 1 which happens at most $\lfloor \frac{n}{2} \rfloor$ times $\sim n$ pieces of info

so $\mathcal{A}$ must make at least

$$\frac{n}{2} + (2n-2-n) = \frac{n}{2} + n - 2 = \frac{3n}{2} - 2 \quad n=2k$$

n odd it gives $\frac{3n}{2} - \frac{3}{2}$:

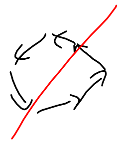
$n = 2k+1$ so at most k comp give 2 pieces in total $2k$ info from these

remains $(2n-2-2k) = (2(2k+1)-2) - 2k = 2k$ pieces

so $\mathcal{A}$ must make at least $k + 2k = 3k$ comp

$$3k = 3 \cdot \left(\frac{n-1}{2}\right) = \frac{3n}{2} - \frac{3}{2}.$$

<u>acyclic</u> digraph: no directed cycle



an <u>acyclic order</u> of a digraph is an ordering $v_1, v_2 \ldots\ldots v_n$ of V s.t. no arc $v_j v_i$ with $j > i$

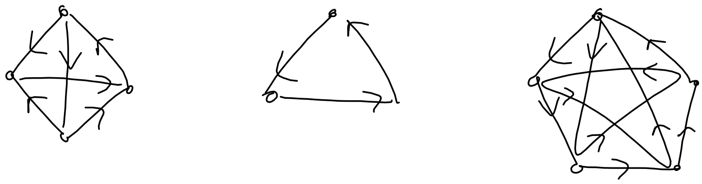
all arcs forward

<u>Lemma</u> a digraph D is acyclic if and only if it has an acyclic order.

exercise.

A tournament is an orientation of a complete graph K_n

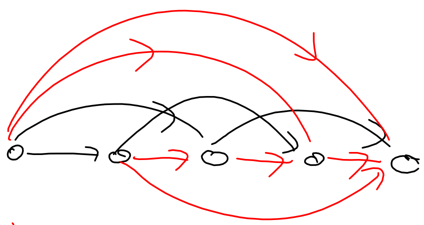
ex



There is only one acyclic tournament on n vertices: transitive tournament TT_n

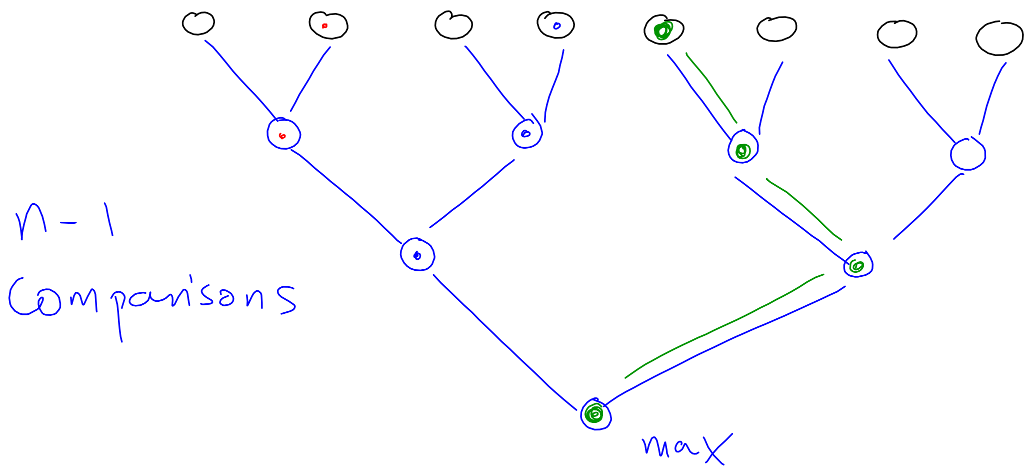
Adversary orients edges of K_n n = no of numbers so that the oriented edges form an acyclic digraph.

<u>Lemma</u> If $D=(V,A)$ is acyclic on n vertices then we can add arcs A' s.t. $D+A'$ is the transitive tournament TT_n



1 2 3 4 5

max and second largest element via tourment method.



Where is the second largest?

only candidates are those who only lost to max so at most $\log_2 n$ (assume $n=2^k$)

so $\leq \log_2 n$ candidates for second largest

$n \xrightarrow[\text{comp}]{n-1} \cdot$ max

$\log_2 n \ (= k)$

$\downarrow$ $\log_2 n - 1$ comp

$\cdot$ second largest

total $n - 1 + \log_2 n - 1$

$= n + \log_2 n - 2$ comp.

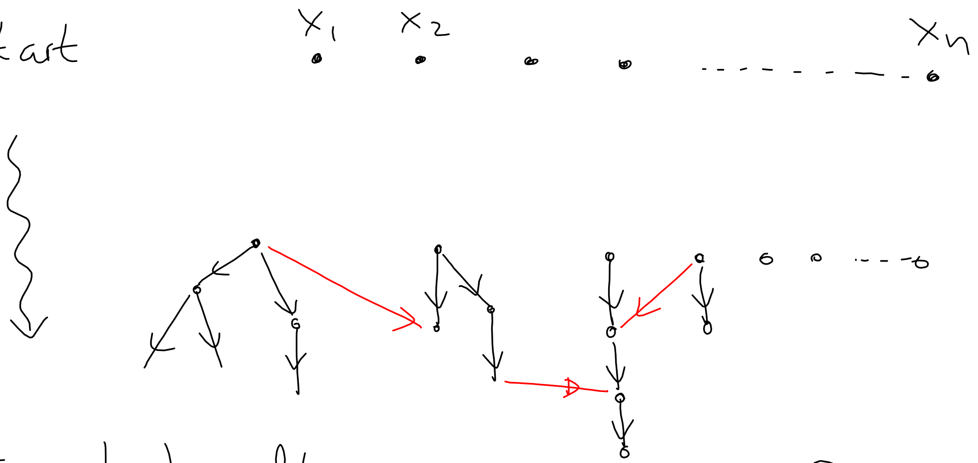
NB we look min and second smallest

Adversary can force this # of comp:

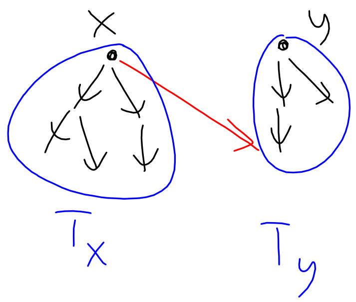
Adversary wants to keep many candidates for second largest

terminology: $\bullet \longrightarrow \circ$ usefull for max

$\circ \longrightarrow \circ$ potentially usefull for second largest start

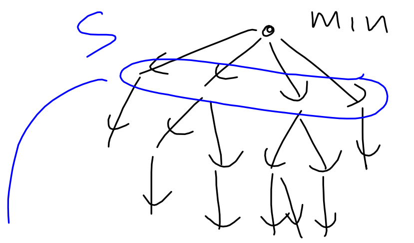


How to handle compare x and y?



add $x \to y$ because $|T_x| \geq |T_y|$ (*)

When algorithm terminates:



$n-1$ black edges

Candidates for second smallest.

at least $\log_2 n$ candidates by (*)

any A for min, 2nd min must find info about min in S $\Rightarrow |S|-1$ comp.

<u>So</u> A must use at least

$$n-1 + \log_2 n - 1 = n + \log_2 n - 2 \text{ comp.}$$