Selection and Adversary Arguments

Note Title 3/15/2012

Selection and Adversary Arguments

Note Title 3/15/2012

1. Finding max and min among n keys

Selection and Adversary Arguments

Note Title 3/15/2012

1. Finding max and min among n keys

A brute force solution:

Selection and Adversary Arguments

Note Title 3/15/2012

1. Finding max and min among n keys

A brute force solution:

a. Find max among n keys

Selection and Adversary Arguments

Note Title 3/15/2012

1. Finding max and min among n keys

A brute force solution:

a. Find max among n keys

this takes $n-1$ comparisons

Selection and Adversary Arguments

Note Title 3/15/2012

1. Finding max and min among n keys

A brute force solution:

a. Find max among n keys

this takes $n-1$ comparisons

b. Find min among remaining $n-1$ keys

Selection and Adversary Arguments

Note Title 3/15/2012

1. Finding max and min among n keys

A brute force solution:

a. Find max among n keys

this takes $n-1$ comparisons

b. Find min among remaining $n-1$ keys

this takes $n-2$ comparisons

Selection and Adversary Arguments

Note Title 3/15/2012

1. Finding max and min among n keys

A brute force solution:

a. Find max among n keys

this takes $n-1$ comparisons

b. Find min among remaining $n-1$ keys

this takes $n-2$ comparisons

Total: $(n-1)+(n-2) = 2n-3$ comparisons

We can do better than that!

Case of even n:

$a_1\ a_2$ $a_3\ a_4$ $a_5\ a_6$ $a_{n-1}\ a_n$

All n contestants are grouped in pairs

Case of even n:

a_1 a_2	a_3 a_4	a_5 a_6	a_{n-1} a_n

All n contestants are grouped in pairs

Compare them in these pairs

Case of even n:

max_1	max_2	max_3	$max_{n/2}$
a_1 a_2	a_3 a_4	a_5 a_6	a_{n-1} a_n
min_1	min_2	min_3	$min_{n/2}$

All n contestants are grouped in pairs

Compare them in these pairs

Case of even n: $\frac{n}{2}$ comparisons so far

max_1	max_2	max_3		$max_{n/2}$
a_1 a_2	a_3 a_4	a_5 a_6		a_{n-1} a_n
min_1	min_2	min_3		$min_{n/2}$

All n contestants are grouped in pairs

Compare them in these pairs

Case of even n:

$\frac{n}{2}$ comparisons so far

max_1	max_2	max_3	$max_{n/2}$
a_1 a_2	a_3 a_4	a_5 a_6	a_{n-1} a_n
min_1	min_2	min_3	$min_{n/2}$

Select Max of $\frac{n}{2}$ max_i's

Case of even n: $\frac{n}{2}$ comparisons so far

max_1	max_2	max_3		$max_{n/2}$ $\frac{n}{2} - 1$ comparisons
a_1 a_2	a_3 a_4	a_5 a_6		a_{n-1} a_n
min_1	min_2	min_3		$min_{n/2}$

Select Max of $\frac{n}{2}$ max_i's

Case of even n:

$\frac{n}{2}$ comparisons so far

max_1	max_2	max_3	$max_{n/2}$
a_1 a_2	a_3 a_4	a_5 a_6	a_{n-1} a_n
min_1	min_2	min_3	$min_{n/2}$

$\frac{n}{2} - 1$ comparisons

select Min of $\frac{n}{2}$ min_i's

Case of even n: $\frac{n}{2}$ comparisons so far

max_1	max_2	max_3		$max_{n/2}$	$\frac{n}{2} - 1$ comparisons
a_1 a_2	a_3 a_4	a_5 a_6		a_{n-1} a_n	
min_1	min_2	min_3		$min_{n/2}$	$\frac{n}{2} - 1$ comparisons

select Min of $\frac{n}{2}$ min_i's

Case of even n: $\frac{n}{2}$ comparisons so far

max_1	max_2	max_3	$max_{n/2}$	$\frac{n}{2} - 1$ comparisons
a_1 a_2	a_3 a_4	a_5 a_6	a_{n-1} a_n	
min_1	min_2	min_3	$min_{n/2}$	$\frac{n}{2} - 1$ comparisons

Total # of comparisons $= \frac{n}{2} + \frac{n}{2} - 1 + \frac{n}{2} - 1$

integer

$$= \frac{3}{2}n - 2 = \left\lceil \frac{3}{2}n \right\rceil - 2$$

$= \frac{3}{2} n - 2 \quad = \left\lceil \frac{3}{2} n \right\rceil - 2$

There are equal because n is even so $\frac{n}{2}$ is an integer, and so is $3 \frac{n}{2} = \frac{3}{2} n$

Case of odd n:

a_1 a_2	a_3 a_4	a_5 a_6	a_{n-2} a_{n-1}	a_n

All n contestants are grouped in pairs

Case of odd n:

$a_1\ a_2$ $\quad$ $a_3\ a_4$ $\quad$ $a_5\ a_6$ $\qquad\qquad$ $a_{n-2}\ a_{n-1}$ $\quad$ a_n

All n contestants are grouped in pairs
except for the last one

Case of odd n:

a_1 a_2 | a_3 a_4 | a_5 a_6 | a_{n-2} a_{n-1} | a_n

Compare them in pairs

Case of odd n:

max_1	max_2	max_3		$max_{(n-1)/2}$	a_n
a_1 a_2	a_3 a_4	a_5 a_6		a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3		$min_{(n-1)/2}$	a_n

Compare them in pairs

Case of odd n:

max_1	max_2	max_3		$max_{(n-1)/2}$	a_n
a_1 a_2	a_3 a_4	a_5 a_6		a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3		$min_{(n-1)/2}$	a_n

Compare them in pairs

This took $\frac{n-1}{2}$ comparisons

Case of odd n:

$\frac{n-1}{2}$ comparisons so far

max_1	max_2	max_3		$max_{(n-1)/2}$	a_n
a_1 a_2	a_3 a_4	a_5 a_6		a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3		$min_{(n-1)/2}$	a_n

Compare them in pairs

Case of odd n: $\frac{n-1}{2}$ comparisons so far

max_1	max_2	max_3	$max_{(n-1)/2}$	a_n
$a_1\ a_2$	$a_3\ a_4$	$a_5\ a_6$	$a_{n-2}\ a_{n-1}$	a_n
min_1	min_2	min_3	$min_{(n-1)/2}$	a_n

Select Max of max_i's and a_n

Case of odd n: $\frac{n-1}{2}$ comparisons so far

max_1	max_2	max_3	$max_{(n-1)/2}$	a_n
a_1 a_2	a_3 a_4	a_5 a_6	a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3	$min_{(n-1)/2}$	a_n

Select Max of max_i's and a_n

This took $\frac{n-1}{2}$ comparisons

Case of odd n: $\frac{n-1}{2}$ comparisons so far

max_1	max_2	max_3	$max_{(n-1)/2}$	a_n
a_1 a_2	a_3 a_4	a_5 a_6	a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3	$min_{(n-1)/2}$	a_n

$\frac{n-1}{2}$ comparisons

Select Max of max_i's and a_n

Case of odd n:

$\frac{n-1}{2}$ comparisons so far

max_1	max_2	max_3	$max_{(n-1)/2}$	a_n
a_1 a_2	a_3 a_4	a_5 a_6	a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3	$min_{(n-1)/2}$	a_n

$\frac{n-1}{2}$ comparisons

Select Min of min_i's and a_n

Case of odd n: $\frac{n-1}{2}$ comparisons so far

max_1	max_2	max_3		$max_{(n-1)/2}$	a_n $\frac{n-1}{2}$ comparisons
a_1 a_2	a_3 a_4	a_5 a_6		a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3		$min_{(n-1)/2}$	a_n

Select Min of min_i's and a_n

This took $\frac{n-1}{2}$ comparisons

Case of odd n: $\frac{n-1}{2}$ comparisons so far

max_1	max_2	max_3		$max_{(n-1)/2}$	a_n $\frac{n-1}{2}$ comparisons
a_1 a_2	a_3 a_4	a_5 a_6		a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3		$min_{(n-1)/2}$	a_n $\frac{n-1}{2}$ comparisons

Select Min of min_i's <u>and</u> a_n

Case of odd n: $\frac{n-1}{2}$ comparisons so far

max_1	max_2	max_3	$max_{(n-1)/2}$	a_n $\frac{n-1}{2}$ comparisons
a_1 a_2	a_3 a_4	a_5 a_6	a_{n-2} a_{n-1}	a_n
min_1	min_2	min_3	$min_{(n-1)/2}$	a_n $\frac{n-1}{2}$ comparisons

The total number of comparisons was:

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

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

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

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

$$\left\lceil \frac{3}{2}n \right\rceil - \frac{4}{2}$$

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

$$\left\lceil \frac{3}{2}n \right\rceil - \frac{4}{2}$$

These two are equal because n is odd, and so is $3n$, so $\frac{3}{2}n$ is <u>not</u> integer $\left(= k + \frac{1}{2}\right)$

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

$$\left\lceil \frac{3}{2}n \right\rceil - \frac{4}{2}$$

So, this "rounds" $\frac{3}{2}n$ up

These two are equal because n is odd, and so is $3n$, so $\frac{3}{2}n$ is not integer $\left(= k + \frac{1}{2}\right)$

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

$\lceil \frac{3}{2}n \rceil - \frac{4}{2}$

So, this "rounds" $\frac{3}{2}n$ up

And so does this

These two are equal because n is odd, and so is $3n$, so $\frac{3}{2}n$ is <u>not</u> integer $(= k + \frac{1}{2})$

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

$\left\lceil \frac{3}{2}n \right\rceil - \frac{4}{2}$

So, this "rounds" $\frac{3}{2}n$ up

$= \left\lceil \frac{3}{2}n \right\rceil - 2$

And so does this

These two are equal because n is odd, and so is $3n$, so $\frac{3}{2}n$ is <u>not</u> integer $\left(= k + \frac{1}{2}\right)$

Conclusion

The above algorithm finds Max and Min among n keys in no more than $\lceil \frac{3}{2} n \rceil - 2$ comparisons.

Conclusion

The above algorithm finds Max and Min among n keys in no more than $\lceil \frac{3}{2} n \rceil - 2$ comparisons.

We will show that no other algorithm that finds Max and Min by comparisons of keys can do less comparisons in the worst case.

Adversary strategy for finding Max & Min

Adversary strategy for finding Max & Min

The adversary will construct a worst-case scenario for any algorithm in the above mentioned class

Adversary strategy for finding Max & Min

The adversary will construct a worst-case scenario for any algorithm in the above mentioned class.

It will do it while the said algorithm is running.

Adversary strategy for finding Max & Min

If x is max then every other key,
except x lost at least 1 comparison

Adversary strategy for finding Max & Min

If x is max then every other key, except x lost at least 1 comparison

Knowing that is 1 unit of information

Adversary Strategy for Finding Max & Min

If x is max then every other key, except x lost at least 1 comparison

Knowing that is 1 unit of information

So, $n-1$ units of information are needed to know that x is max.

Adversary Strategy for Finding Max & Min

If y is min then every other key, except y won at least 1 comparison

Knowing that is 1 unit of information

So, $n-1$ units of information are needed to know that y is min.

Adversary strategy for finding Max & Min

In total, $n-1 + n-1 = 2n-2$

unit of information are needed

Adversary strategy for finding Max & Min

In total, $n-1 + n-1 = 2n-2$ unit of information are needed

in order to know for sure that x is max and y is min

Adversary strategy for finding Max & Min

The adversary will construct the set of numbers under consideration

Adversary strategy for finding Max & Min

The adversary will construct the set of numbers under consideration, so that each time the algorithm in question makes a comparison,

Adversary strategy for finding Max & Min

The adversary will construct the set of numbers under consideration, so that each time the algorithm in question makes a comparison, the minimum of units of information is gien away.

Adversary strategy for finding Max & Min

Key status	Meaning
W	Has won at least one comparison and never lost
L	Has lost at least one comparison and never won
WL	Has won and lost at least one comparison
N	Has not yet participated in a comparison

Adversary strategy for finding Max & Min

Status of keys x and y compared by an algorithm	Adversary response	New status	Units of new information
N, N	$x > y$	W, L	2
W, N or WL, N	$x > y$	W, L or WL, L	1
L, N	$x < y$	L, W	1
W, W	$x > y$	W, WL	1
L, L	$x > y$	WL, L	1
W, L or WL, L or W, WL	$x > y$	No change	0
WL, WL	Consistent with assigned values	No change	0

Table 5.1 The adversary strategy for the min and max problem

Adversary strategy for finding Max & Min

2 units of information will be given to no more than $\left\lceil \frac{n}{2} \right\rceil$ comparisons

Adversary strategy for finding Max & Min

2 units of information will be given to no more than $\left\lfloor \frac{n}{2} \right\rfloor$ comparisons

This will give in total $2 \cdot \left\lfloor \frac{n}{2} \right\rfloor$ units of information.

Adversary strategy for finding Max & Min

2 units of information will be given to no more than $\lfloor \frac{n}{2} \rfloor$ comparisons

This will give in total $2 \cdot \lfloor \frac{n}{2} \rfloor$ units of information.

The remaining needed $2n-2 - 2 \cdot \lfloor \frac{n}{2} \rfloor$ units of information will require $2n-2 - 2 \lfloor \frac{n}{2} \rfloor$ comparisons.

Adversary strategy for finding Max & Min

So, at least

$\lfloor \frac{n}{2} \rfloor + 2n - 2 - 2\lfloor \frac{n}{2} \rfloor$ comparisons

are necessary.

Adversary strategy for finding Max & Min

So, at least

$\cancel{\left\lfloor \frac{n}{2} \right\rfloor} + 2n - 2 - \cancel{2}\left\lfloor \frac{n}{2} \right\rfloor$ comparisons

are necessary.

$= 2n - \left\lfloor \frac{n}{2} \right\rfloor - 2 \approx$

Adversary strategy for finding Max & Min

So, at least

$$\cancel{\left\lfloor \frac{n}{2} \right\rfloor} + 2n - 2 - \cancel{2}\left\lfloor \frac{n}{2} \right\rfloor \quad \text{comparisons}$$

are necessary.

$$= 2n - \left\lfloor \frac{n}{2} \right\rfloor - 2 =$$

$$= n + \left(n - \left\lfloor \frac{n}{2} \right\rfloor\right) - 2 =$$

Adversary strategy for finding Max & Min

So, at least

$\cancel{\lfloor \frac{n}{2} \rfloor} + 2n - 2 - \cancel{2}\lfloor \frac{n}{2} \rfloor$ comparisons

are necessary.

$\lceil \frac{n}{2} \rceil + \lfloor \frac{n}{2} \rfloor = n$

$= 2n - \lfloor \frac{n}{2} \rfloor - 2 =$

$= n + \left(n - \lfloor \frac{n}{2} \rfloor\right) - 2 =$

Adversary Strategy for Finding Max & Min

So, at least

$\cancel{\lfloor \frac{n}{2} \rfloor} + 2n - 2 - \cancel{2}\lfloor \frac{n}{2} \rfloor$ comparisons

are necessary.

$= 2n - \lfloor \frac{n}{2} \rfloor - 2 =$

$= n + \left(n - \lfloor \frac{n}{2} \rfloor\right) - 2 =$

$\lceil \frac{n}{2} \rceil + \lfloor \frac{n}{2} \rfloor = n$

so $n - \lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil$

Adversary Strategy for Finding Max & Min

So, at least

$$\cancel{\left\lfloor \frac{n}{2} \right\rfloor} + 2n - 2 - \cancel{2}\left\lfloor \frac{n}{2} \right\rfloor \quad \text{comparisons}$$

are necessary.

$$= 2n - \left\lfloor \frac{n}{2} \right\rfloor - 2 =$$

$$= n + \left(n - \left\lfloor \frac{n}{2} \right\rfloor\right) - 2 = n + \left\lceil \frac{n}{2} \right\rceil - 2 =$$

$$\left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor = n$$

so $n - \left\lfloor \frac{n}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil$

Adversary strategy for finding Max & Min

So, at least

$$\cancel{\left\lfloor \frac{n}{2} \right\rfloor} + 2n - 2 - \cancel{2}\left\lfloor \frac{n}{2} \right\rfloor \quad \text{comparisons}$$

are necessary.

$$= 2n - \left\lfloor \frac{n}{2} \right\rfloor - 2 =$$

$$\left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor = n$$

so $n - \left\lfloor \frac{n}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil$

$$= n + \left(n - \left\lfloor \frac{n}{2} \right\rfloor\right) - 2 = n + \left\lceil \frac{n}{2} \right\rceil - 2 =$$

$$= \left\lceil \frac{n}{2} + n \right\rceil - 2 = \left\lceil \frac{3}{2} n \right\rceil - 2.$$

The Tournament Method for finding the second-largest key

The Tournament Method for finding the second-largest key

It is easy to find second largest in $n-1$ (to know the largest)

The Tournament Method for finding the second-largest key

It is easy to find second largest in $n-1$ (to know the largest) plus

$n-2$ (to find the largest in the remaining $n-1$ elements)

The Tournament Method for finding the second-largest key

It is easy to find second largest in

$n-1$ (to know the largest) plus

$n-2$ (to find the largest in the remaining $n-1$ elements)

for a total of

$2n-3$ comparisons

The Tournament Method for finding the second-largest key

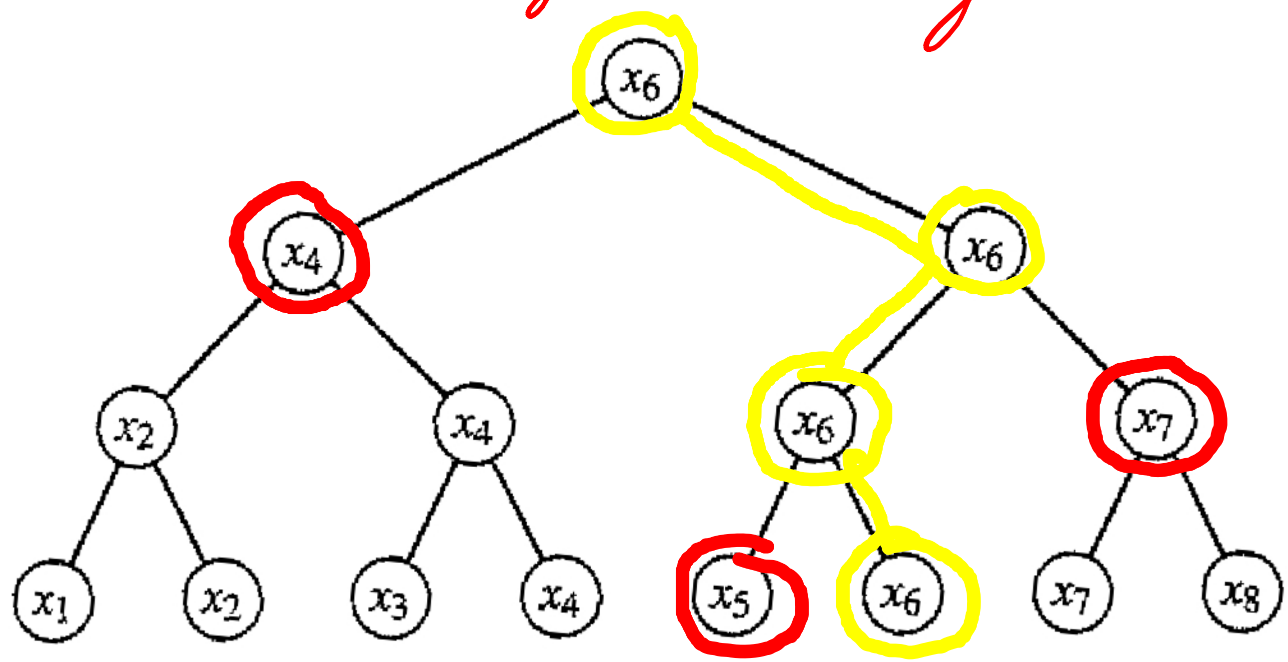


Figure 5.1 An example of a tournament; max $= x_6$; secondLargest may be x_4, x_5, or x_7.

Number of comparisons:

Number of comparisons:

$n - 1$ to find the winner of the tournament

Number of comparisons:

$n - 1$ to find the winner of the tournament

(no surprises here)

Number of comparisons:

$n - 1$ to find the winner of the tournament

(no surprises here)

plus

$\lceil \lg n \rceil - 1$ to find the largest among

$\lceil \lg n \rceil$ losers to the winner.

Number of comparisons:

$n-1$ to find the winner of top tournament

(no surprises here)

plus

$\lceil \lg n \rceil - 1$ to find the largest among

$\lceil \lg n \rceil$ losers to the winer.

Total # comps: $n + \lceil \lg n \rceil - 2$

Adversary for second largest

After any algorithm that finds it by comparisons of keys halts, there is only 1 key that haven't lost. For otherwise, if there were 2 (or more) each of those could be the 2nd largest.

So, there must be $n-1$ losers and 1 who never lost, call it max.

Let K be the number of those who lost to the max.

Of those K losers, K-1 must have lost more than once, or else if they were 2 or more that lost only to the max, one cannot tell which of those is the 2nd largest.

In order to make $n-1$ losers and $k-1$ double losers, at least

$n-1+k-1 = n+k-2$ comparisons are needed.

We will devise an adversary strategy for any algorithm that finds 2nd largest key by means of comparisons of keys to result in $k \geq \lceil \lg n \rceil$.

Adversary for second largest

Initial situation: $w(x) = 1$ for every key x

Case	Question: compare x to y	Adversary reply	Updating of weights
$w(x) > w(y)$		$x > y$	New $w(x) = \text{prior } (w(x) + w(y))$; new $w(y) = 0$. The winner takes all
$w(x) = w(y) > 0$		Same as above.	Same as above.
$w(y) > w(x)$		$y > x$	New $w(y) = \text{prior } (w(x) + w(y))$; new $w(x) = 0$.
$w(x) = w(y) = 0$		Consistent with previous replies.	No change.

Lemma. The above adversary strategy will force comparisons of the max to $\lceil \lg n \rceil$ distinct keys.

__Proof.__ Let K be the number of distinct keys that lost to the max in this order:

$$y_1 \quad y_2 \quad y_3 \quad \cdots \quad y_K.$$

Let w_i be the "cash" the max accumulated after winning with y_i, $i = 1, 2, 3, \ldots, K$.

$$w_0 = 1$$

— initial "cash"

$$w_k \leq 2w_{k-1} \text{ for } k > 0$$

— no more than doubles his "cash" after each win

$$n = w_K \leq 2^K w_0 = 2^K$$

— "cash" after all wins where K is the number of keys that lost to the max.

$$K \geq \lg n$$

$$K \geq \lceil \lg n \rceil$$

So, the max played with at least $\lceil \lg n \rceil$ other keys.

So, the max won with at least $\lceil \lg n \rceil$ keys.

This completes the proof of the lemma. $\square$

Now, since the number of comparisons is at least $n+k-2$ and $k \geq \lceil \lg n \rceil$, we conclude that the number of comparisons is at least $n + \lceil \lg n \rceil - 2$.

This establishes a lower bound on the worst-case number of comparisons while finding 2nd largest key.

Finding median

Median of n elements
can be found in less than

$$3n + o(n)$$

comparisons in the worst case
The lower bound is more than

$$2n + o(n)$$

Adversary strategy

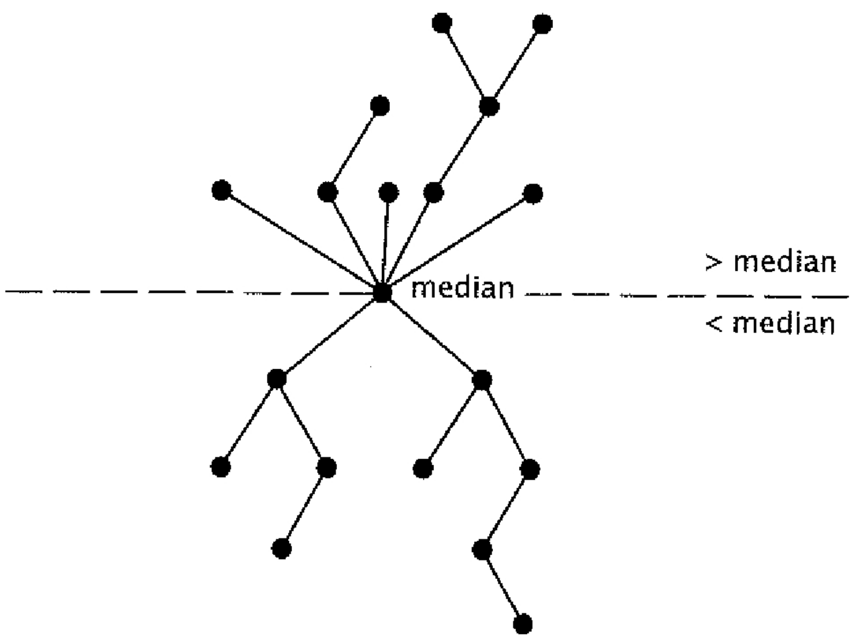


Figure 5.5 Comparisons relating each key to median

Each comparison is represented by an edge in this tree.

Adversary strategy

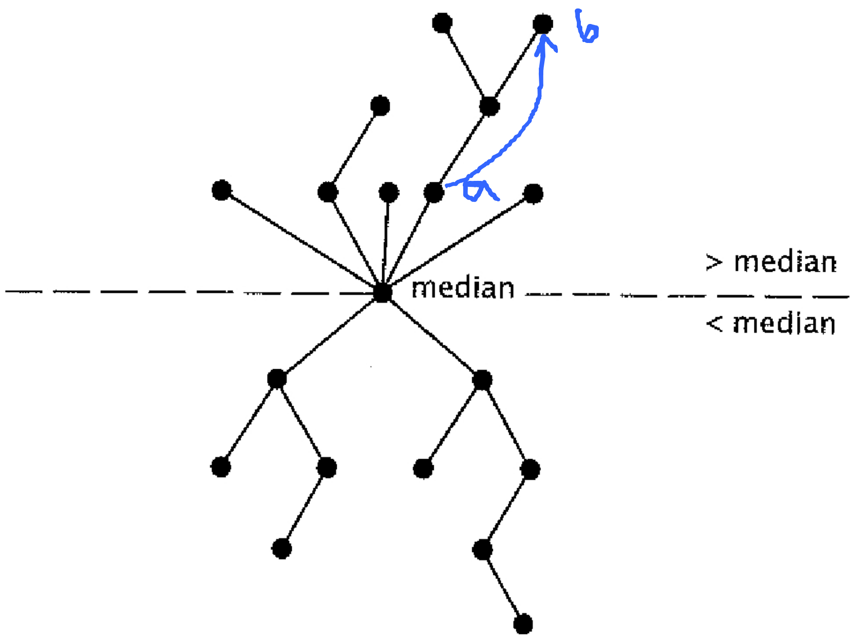


Figure 5.5 Comparisons relating each key to median

Each comparison is represented by an edge in this tree.

An edge → between a and b is a result of comparison of a and b. (b was larger and a was smaller)

Adversary strategy

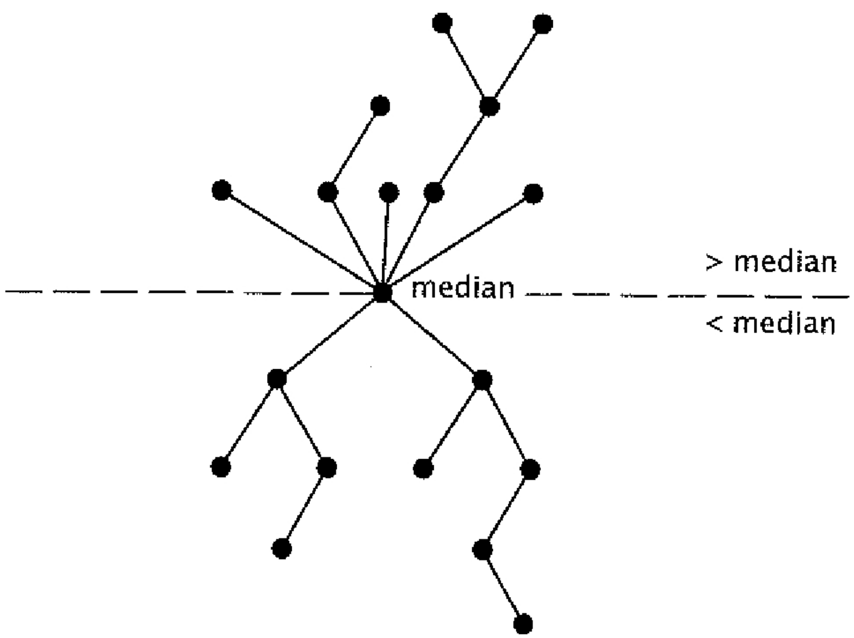


Figure 5.5 Comparisons relating each key to median

Each comparison is represented by an edge in this tree.

Definition. A comparison is crucial if the edge corresponding to it is an edge in this tree. Otherwise, it is not crucial.

Adversary strategy

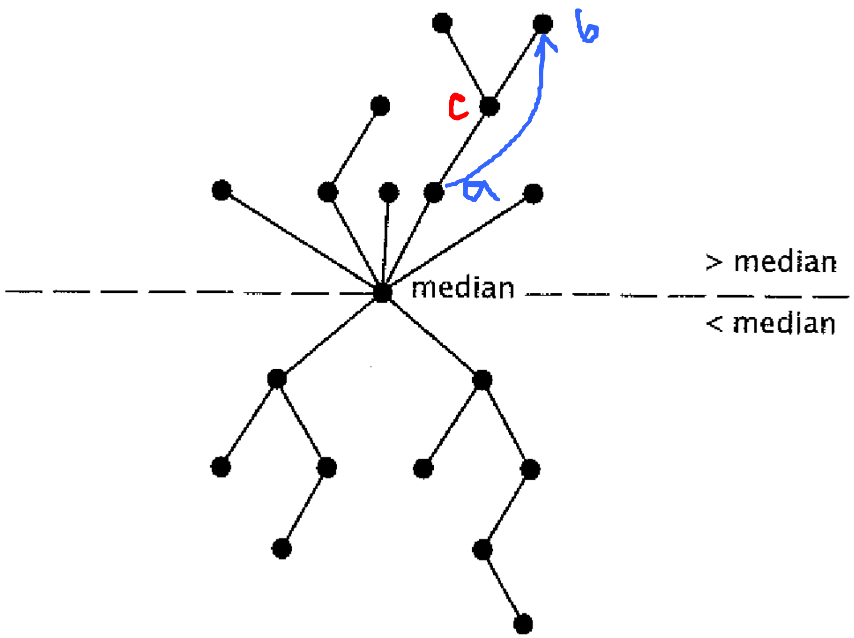


Figure 5.5 Comparisons relating each key to median

Each comparison is represented by an edge in this tree.

Comparison of a and b was non-crucial

Comparisons of a and c, and c and b were both crucial

Adversary strategy

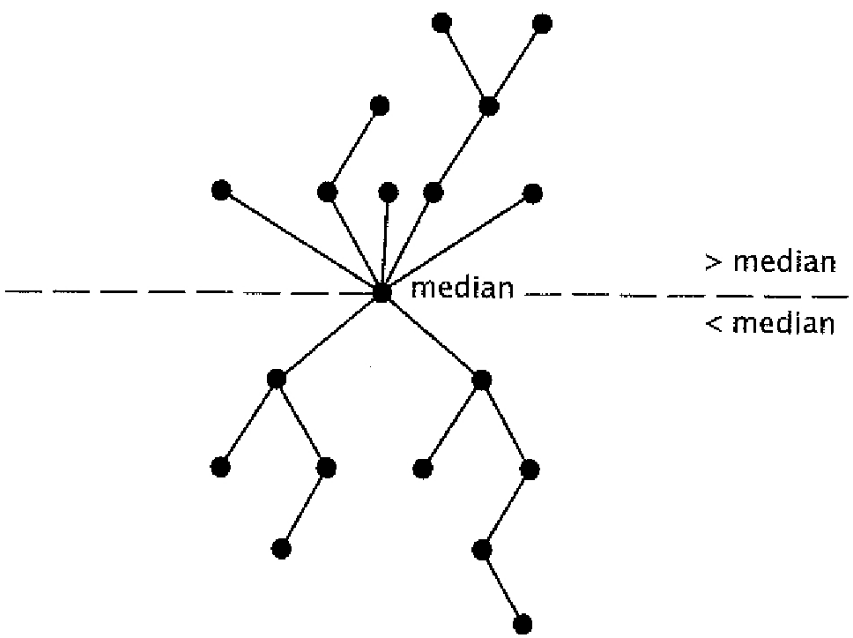


Figure 5.5 Comparisons relating each key to median

Each comparison is represented by an edge in this tree.
A tree with n vertices must have n - 1 edges.
So, only n-1 comparisons are crucial. All others are non-crucial.

Adversary strategy

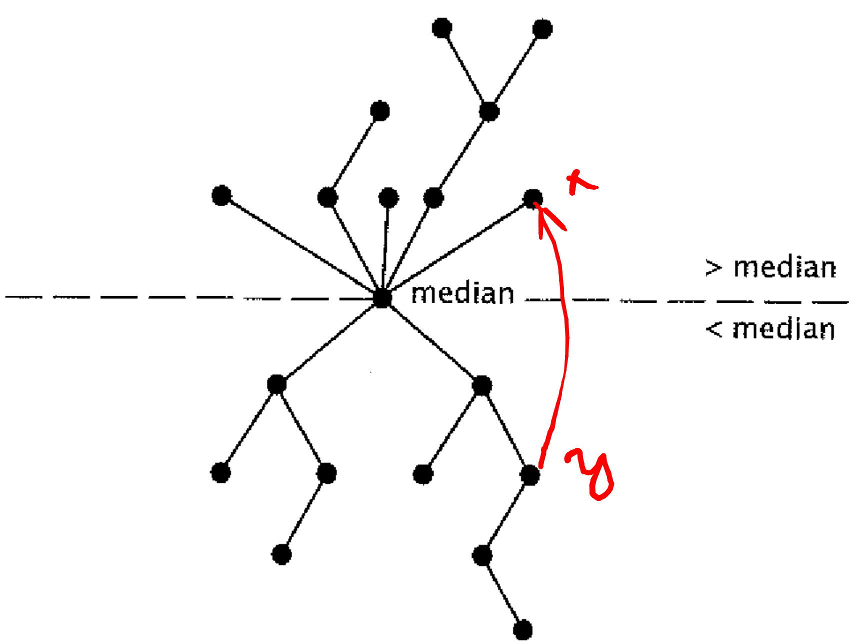


Figure 5.5 Comparisons relating each key to median

Each comparison is represented by an edge in this tree.

A tree with n vertices must have $n-1$ edges.

So, only $n-1$ comparisons are crucial. All others are non-crucial.

All comparisons of x and y ($y <$ median $< x$) are non-crucial

Adversary strategy

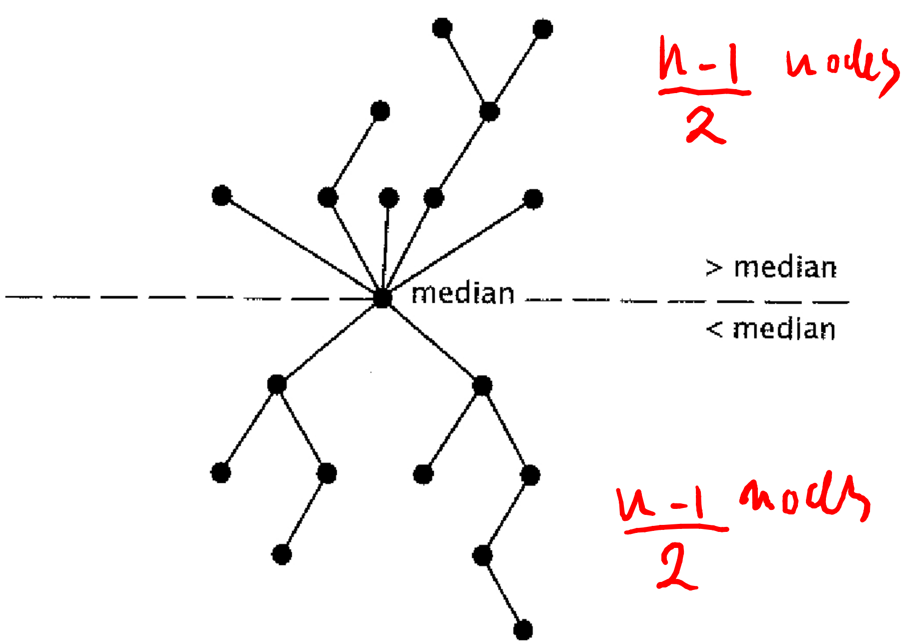
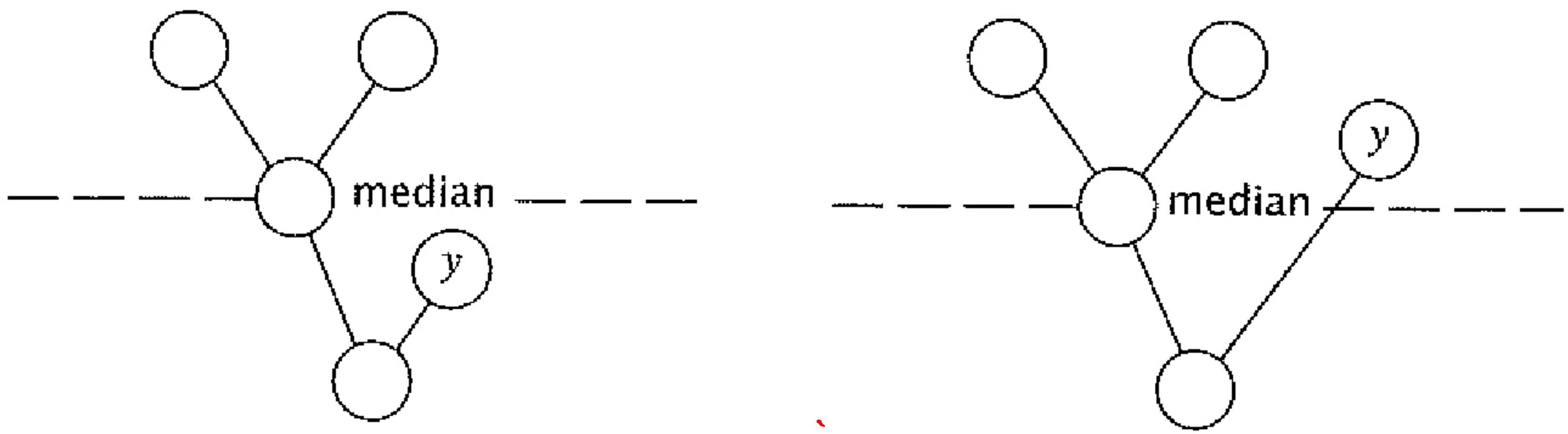


Figure 5.5 Comparisons relating each key to median

Each comparison is represented by an edge in this tree. A tree with n vertices must have $n-1$ edges. So, only $n-1$ comparisons are crucial. All others are non-crucial.

So, adversary can force $\frac{n-1}{2}$ non-crucial comparisons.

This example shows that if the tree of relationships does not establish relationship between a node y and the median then the algorithm is not done.



(a) $y <$ median.

(b) $y >$ median; median is not the median.

Figure 5.6 An adversary conquers a bad algorithm

Comparands	Adversary's action
N, N	Make one key larger than median, the other smaller.
L, N or N, L	Assign a value smaller than median to the key with status N.
S, N or N, S	Assign a value larger than median to the key with status N.

Table 5.4 The adversary strategy for the median-finding problem

All the above comparisons are non-crucial

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons.

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons. Also, the algorithm to correctly find the median, must have performed $n-1$ crucial comparisons.

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons. Also, the algorithm to correctly find the median, must have performed $n-1$ crucial comparisons.

So, a total of at least

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

comparison is needed.

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons. Also, the algorithm to correctly find the median, must have performed $n-1$ crucial comparisons.

So, a total of at least

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

or at least $\lceil \frac{3}{2}n \rceil - 2$

comparison is needed.

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons. Also, the algorithm to correctly find the median, must have performed $n-1$ crucial comparisons.

So, a total of at least

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

or at least

$$\left\lceil \frac{3}{2}n \right\rceil - 2$$

for odd n.

comparison is needed.

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons. Also, the algorithm to correctly find the median, must have performed $n-1$ crucial comparisons.

So, a total of at least

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

or at least $\lceil \frac{3}{2}n \rceil - 2$

for odd n. For even n, it's at least $\lceil \frac{3}{2}n \rceil - 1$

comparison is needed.

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons. Also, the algorithm to correctly find the median, must have performed $n-1$ crucial comparisons.

So, a total of at least

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

or at least $\lceil \frac{3}{2}n \rceil - 2$

for odd n. For even n, it's at least $\lceil \frac{3}{2}n \rceil - 1$, that is, at least $\lceil \frac{3}{2}n \rceil - 2$

comparison is needed.

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons. Also, the algorithm to correctly find the median, must have performed $n-1$ crucial comparisons.

So, a total of at least

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

or at least $\lceil \frac{3}{2} n \rceil - 2$ for odd n. For even n, it's at least $\lceil \frac{3}{2} n \rceil - 1$, that is, at least $\lceil \frac{3}{2} n \rceil - 2$

comparison is needed.

The adversary forced at least $\frac{n-1}{2}$ non-crucial comparisons. Also, the algorithm to correctly find the median, must have performed $n-1$ crucial comparisons.

So, a total of at least

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

comparison is needed.

or at least $\lceil \frac{3}{2}n \rceil - 2$ for odd n. For even n, it's at least $\lceil \frac{3}{2} n \rceil - 1$, that is, at least

Same as the lower bound for finding Min & Max $\lceil \frac{3}{2}n \rceil - 2$