

ALGORITHMS AND DATA STRUCTURES · LECTURE 6

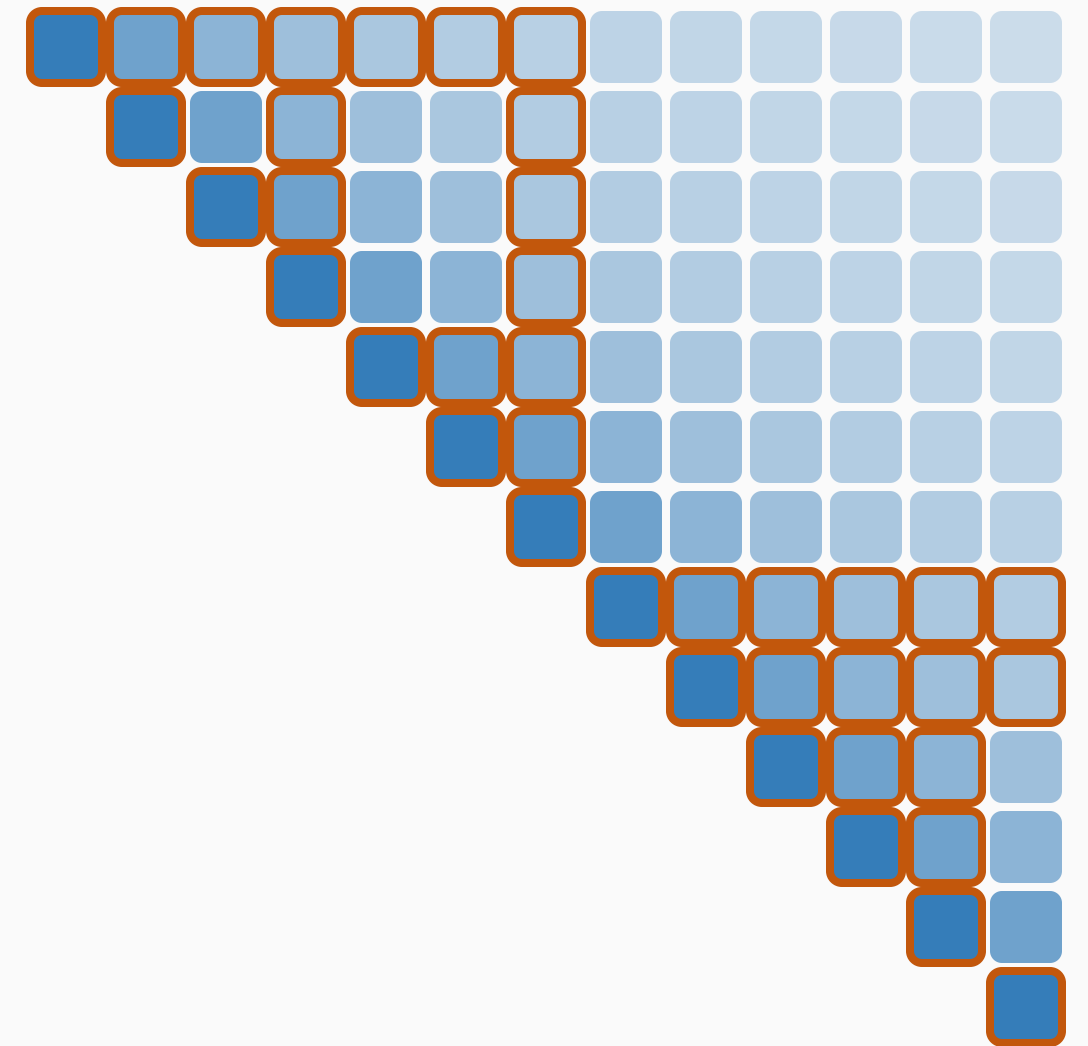
Average-case analysis

Quicksort, and lower bounds for sorting

Heng Guo

School of Informatics, University of Edinburgh

Thursday 8 October 2026 · 16:10–17:00



On 6 October 2026, OpenAI released 722 manuscripts written by internal models, claiming solutions of open research problems. Three are about topics of this course:

	best known before	claimed now
Matrix multiplication (Lecture 4)	$O(n^{2.372})$	$O(n^{2.25+\epsilon})$
Discrete Fourier transform (Lecture 5)	$O(n \log n)$	$O(n(\log n)^{1-10^{-13}})$
Integer multiplication (Lecture 5)	$O(n \log n)$	$O(n(\log n)^{1-2^{-182}})$

Not yet reviewed. OpenAI reports Lean proofs of the matrix bound and of a weaker form of the Fourier result; the integer result is not formalised.

Source: github.com/openai/math, entries 107, 130 and 109.

So far, every running time was a worst-case bound.

Merge sort: $\Theta(n \log n)$ comparisons, always.

Quicksort: $\Theta(n^2)$ in the worst case,
and yet widely used.

? *Why? What is a typical input?*

TODAY

- 1 Typical inputs, and the tools.
- 2 Quicksort: $O(n \log n)$ on average.
- 3 A lower bound for every comparison sort.

comparisons for 1000 numbers

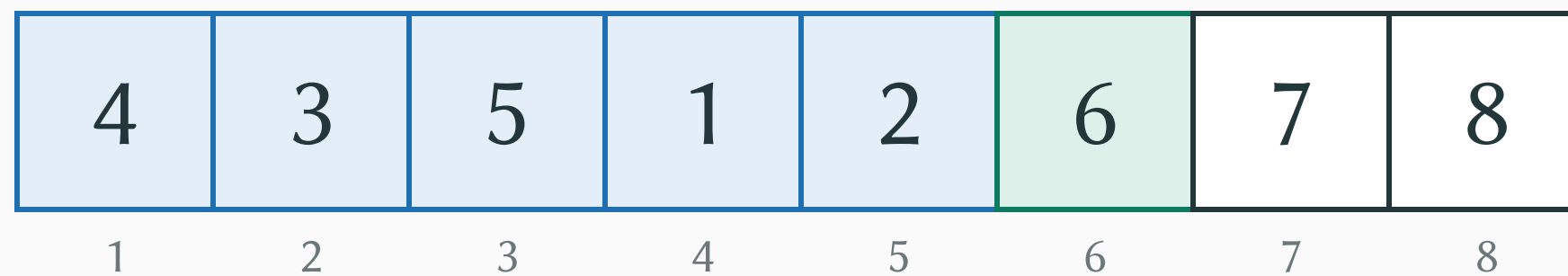
sorted input **499,500**



random order **11,197**



45 times fewer



Swap $A[6]$ and $A[8]$: the pivot 6 is **in its final place**.
7 comparisons in total.

 smaller than the pivot

 $A[j]$, just compared

 larger than the pivot

 the pivot

PARTITION(A, p, r)

1 $x \leftarrow A[r]$ *the pivot*

2 $i \leftarrow p - 1$

3 **for** $j \leftarrow p$ **to** $r - 1$

4 **if** $A[j] < x$

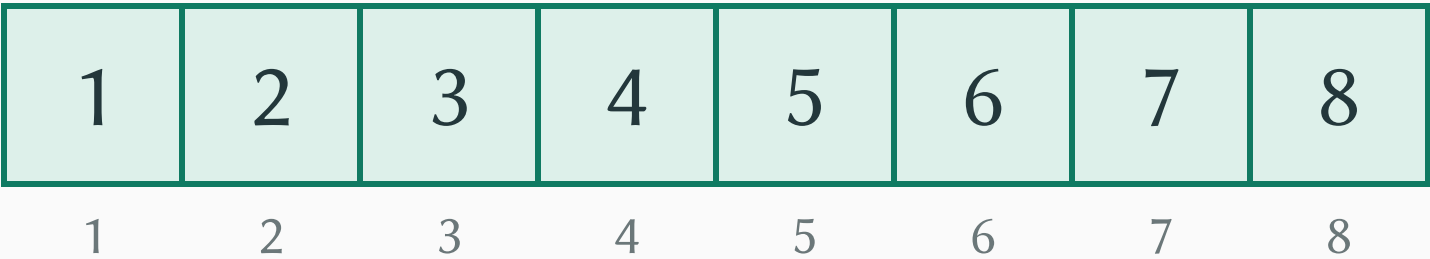
5 $i \leftarrow i + 1$

6 swap $A[i]$ and $A[j]$

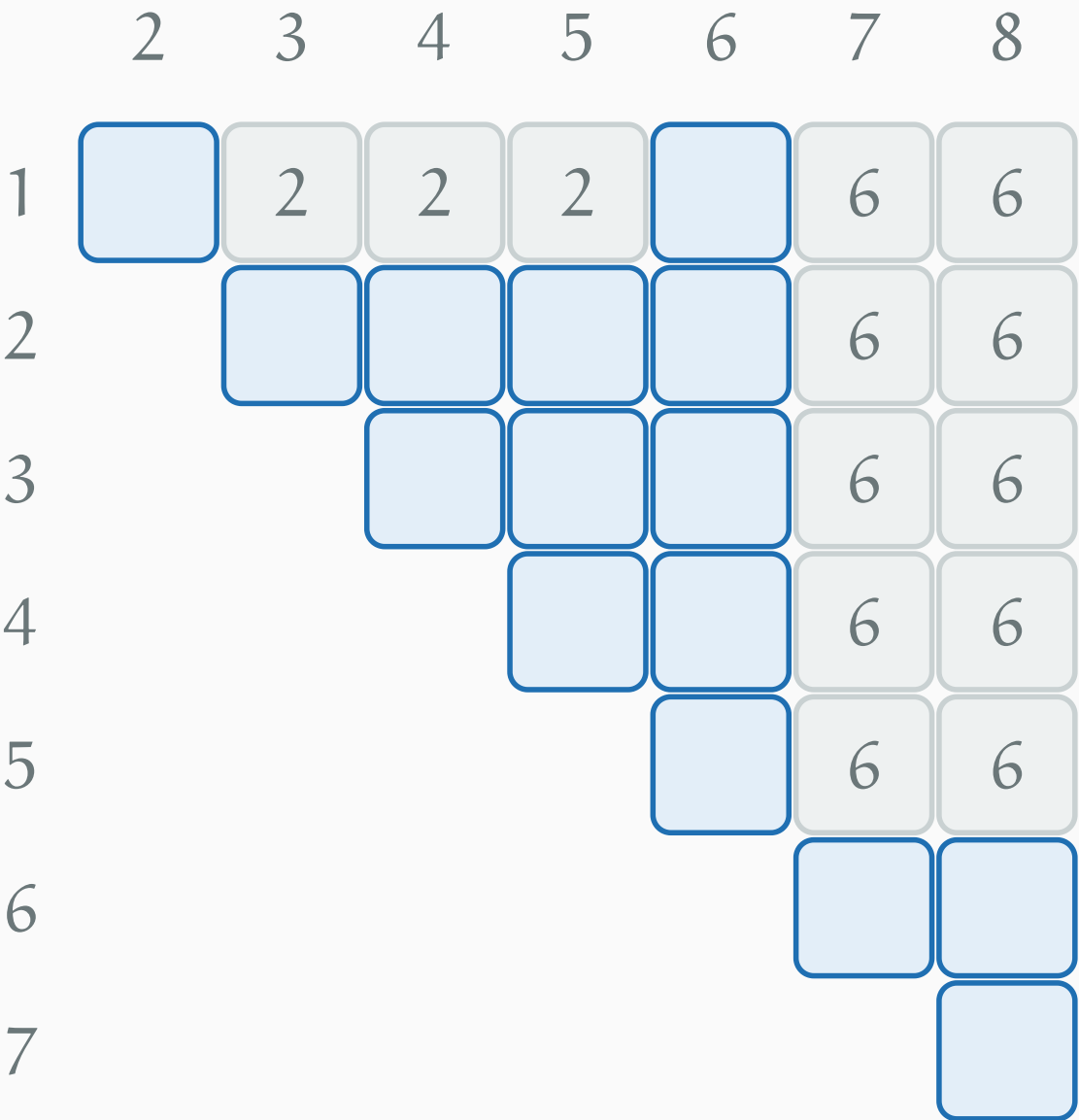
7 swap $A[i + 1]$ and $A[r]$

8 **return** $i + 1$

```
QUICKSORT(A, p, r)
  if p < r
    q ← PARTITION(A, p, r)
    QUICKSORT(A, p, q - 1); QUICKSORT(A, q + 1, r)
```

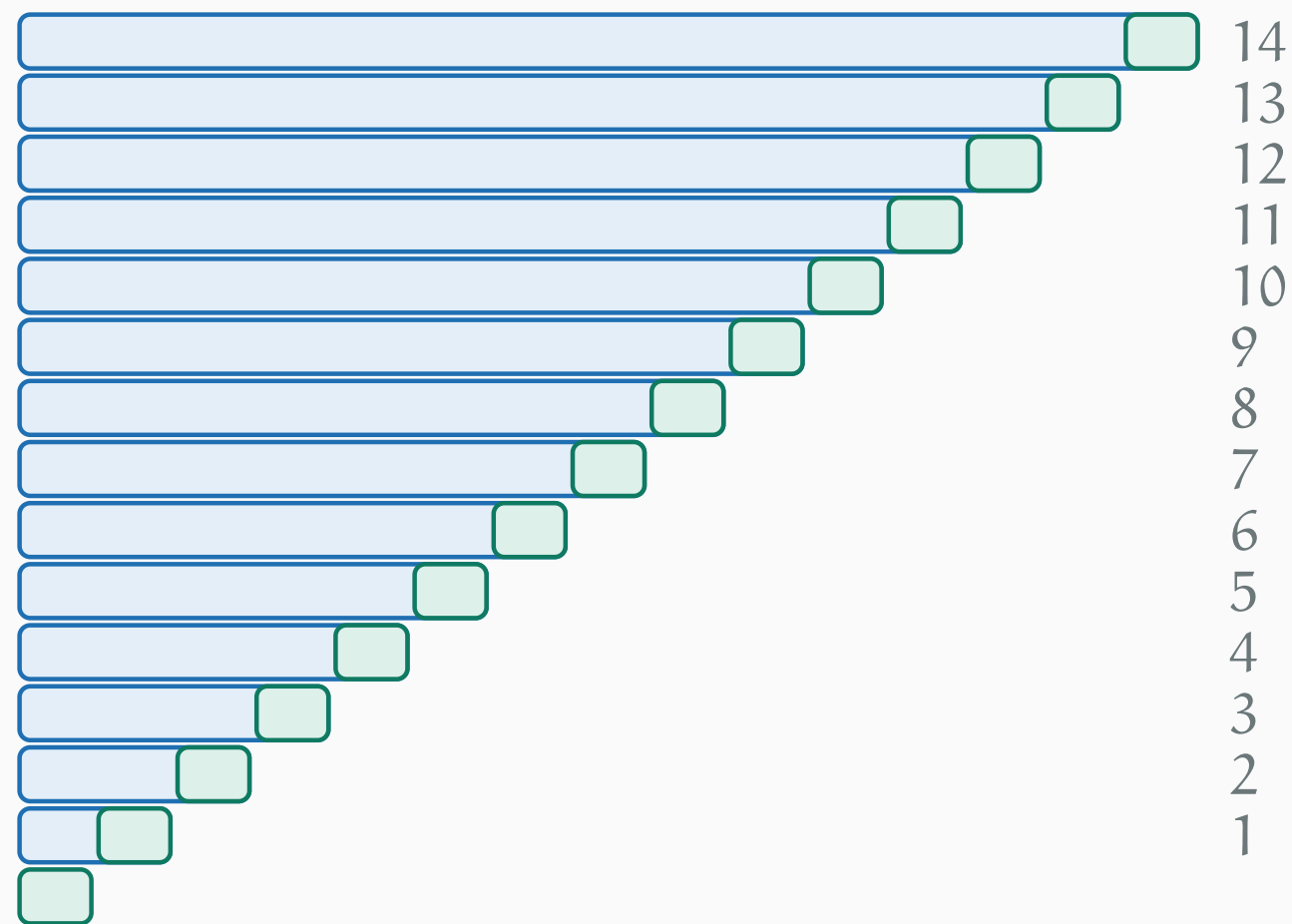


call	pivot	compared with	#
A[1..8]	6	4 7 3 5 1 8 2	7
A[1..5]	2	4 3 5 1	4
A[3..5]	3	5 4	2
A[4..5]	5	4	1
A[7..8]	8	7	1
total			15



15 of the 28 pairs were compared.
Grey: never compared; each shows the pivot that separated the pair.

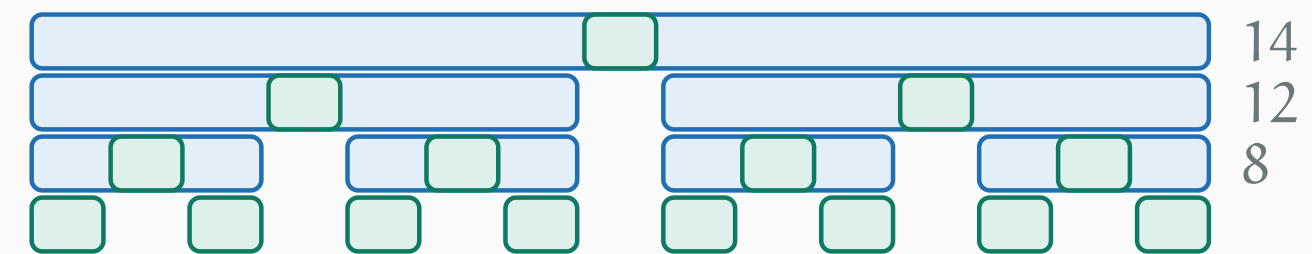
Worst case: the pivot is always the smallest or the largest element.



$$T(n) = T(n-1) + (n-1)$$

$$= \frac{n(n-1)}{2} = \Theta(n^2); \text{ here } 105.$$

Best case: the pivot is always the median.



$$T(n) = 2T\left(\frac{n-1}{2}\right) + (n-1)$$

$$= \Theta(n \log n); \text{ here } 34.$$

 a call of PARTITION on a subarray

 its pivot, now in place

numbers: comparisons on that level

$T(n) = T(n_1) + T(n_2) + (n - 1)$ comparisons,
where $n_1 + n_2 = n - 1$ are the sizes of the two parts.

If every split is 9 : 1:

$$T(n) \leq T\left(\frac{9n}{10}\right) + T\left(\frac{n}{10}\right) + n.$$

? *What does this solve to? Unroll it.*

Each level of the recursion: at most n comparisons.

Longest path: at most $\log_{10/9} n$ levels.
So $T(n) \leq n \log_{10/9} n = O(n \log n)$.

split	levels	$n = 10^6$
1 : 1	$\log_2 n$	20
9 : 1	$\log_{10/9} n$	132
99 : 1	$\log_{100/99} n$	1375

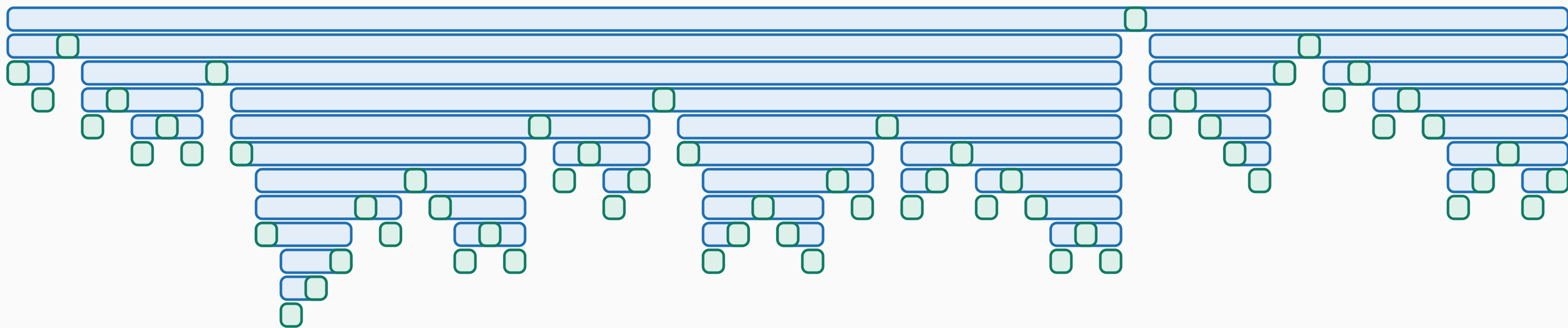
2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---

pivot of rank 2, ..., 9 out of 10:
a split no worse than 9 : 1, with probability 80%.

Worst case the maximum over all inputs of size n (the usual convention)

Best case the minimum

Average case the average over random inputs of size n ; here all $n!$ orderings are equally likely



Random input, $n = 63$: **356** comparisons, 12 levels.

Average over all inputs: 353. Best case: 258. Worst case: 1953.

Expectation of a random variable X :

$$\mathbb{E}[X] = \sum_x x \cdot \Pr[X = x].$$

Indicator variable of an event A :

$X_A = 1$ if A happens, else 0.

$$\mathbb{E}[X_A] = 1 \cdot \Pr[A] + 0 \cdot \Pr[\text{not } A] = \Pr[A].$$

LINEARITY. $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$,
even if X and Y are dependent.

A random permutation π of $\{1, 2, 3\}$;

$X_{ij} = 1$ if $\pi_i > \pi_j$, else 0 ($i < j$).

$X = X_{12} + X_{13} + X_{23}$: the number of **inversions**.

π	X_{12}	X_{13}	X_{23}	X
1 2 3	0	0	0	0
1 3 2	0	0	1	1
2 1 3	1	0	0	1
2 3 1	0	1	1	2
3 1 2	1	1	0	2
3 2 1	1	1	1	3
average	1/2	1/2	1/2	3/2

Each $\mathbb{E}[X_{ij}] = \frac{1}{2}$; for n elements,
 $\mathbb{E}[X] = \binom{n}{2} \cdot \frac{1}{2} = \frac{n(n-1)}{4}$.

QUICKSORT(A, p, r)

```
1  if  $p < r$ 
2   $q \leftarrow \text{PARTITION}(A, p, r)$ 
3  QUICKSORT( $A, p, q - 1$ )
4  QUICKSORT( $A, q + 1, r$ )
```

PARTITION(A, p, r)

```
1   $x \leftarrow A[r]$ 
2   $i \leftarrow p - 1$ 
3  for  $j \leftarrow p$  to  $r - 1$ 
4      if  $A[j] < x$  one comparison
5           $i \leftarrow i + 1$ 
6          swap  $A[i]$  and  $A[j]$ 
7  swap  $A[i + 1]$  and  $A[r]$ 
8  return  $i + 1$ 
```

Calls of **PARTITION**: $\leq n$
each fixes its pivot for good.

Calls of **QUICKSORT**: the first, $\leq 2n + 1$
and two per call of **PARTITION**.

Outside the loop: $O(1)$ per call. $O(n)$

In the loop: one comparison $O(X)$
and $O(1)$ work per iteration.

Running time $O(n + X)$,
where X counts the **comparisons**.

The n elements of A are distinct.

$z_1 < z_2 < \dots < z_n$:

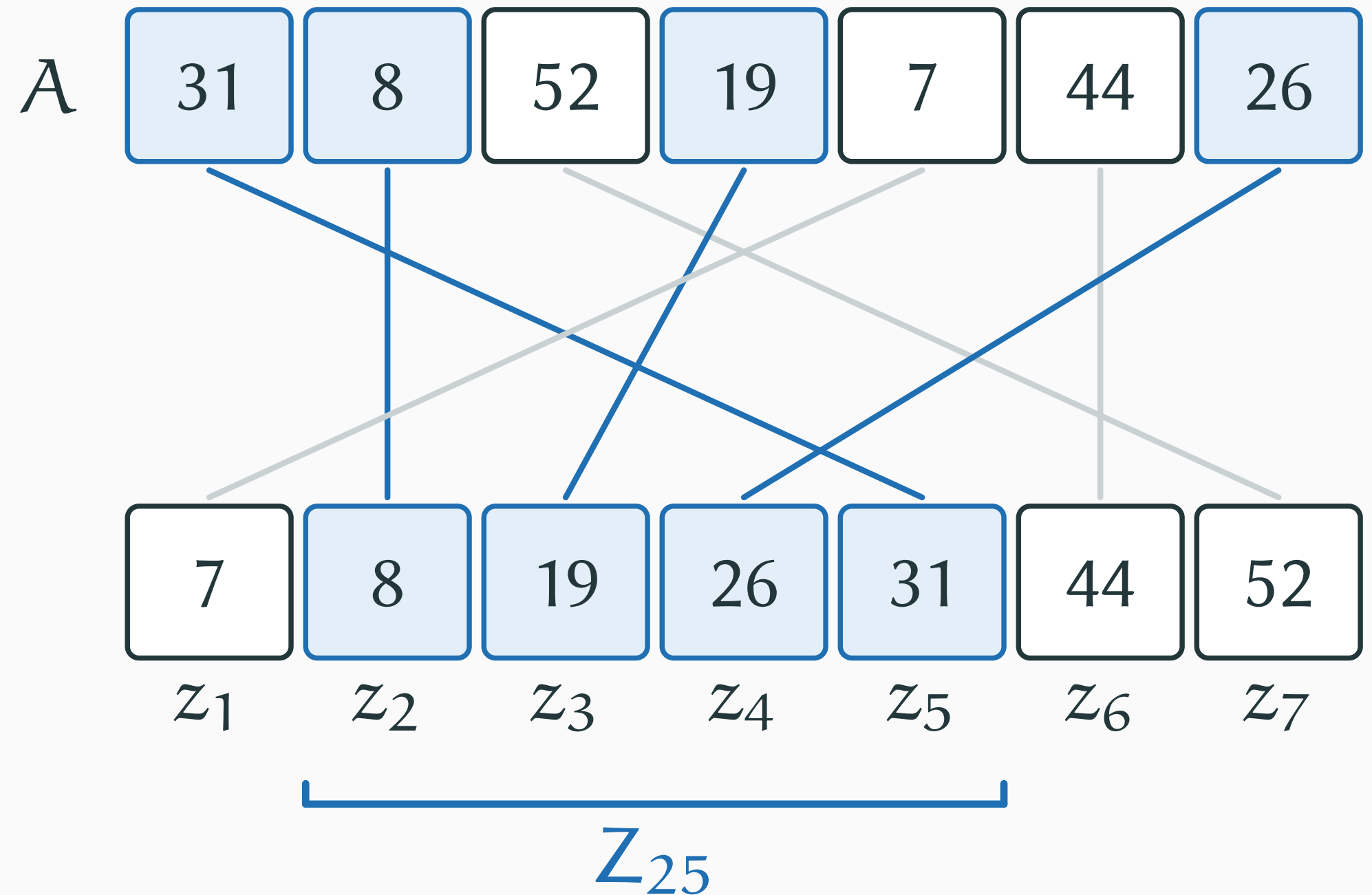
the elements of A **sorted**.

z_i has rank i .

$Z_{ij} = \{z_i, z_{i+1}, \dots, z_j\}$:

the elements of rank i to j .

$X_{ij} = 1$ if z_i and z_j are compared,
else 0 ($i < j$).



EACH PAIR IS COMPARED AT MOST ONCE

AVERAGE CASE OF QUICKSORT

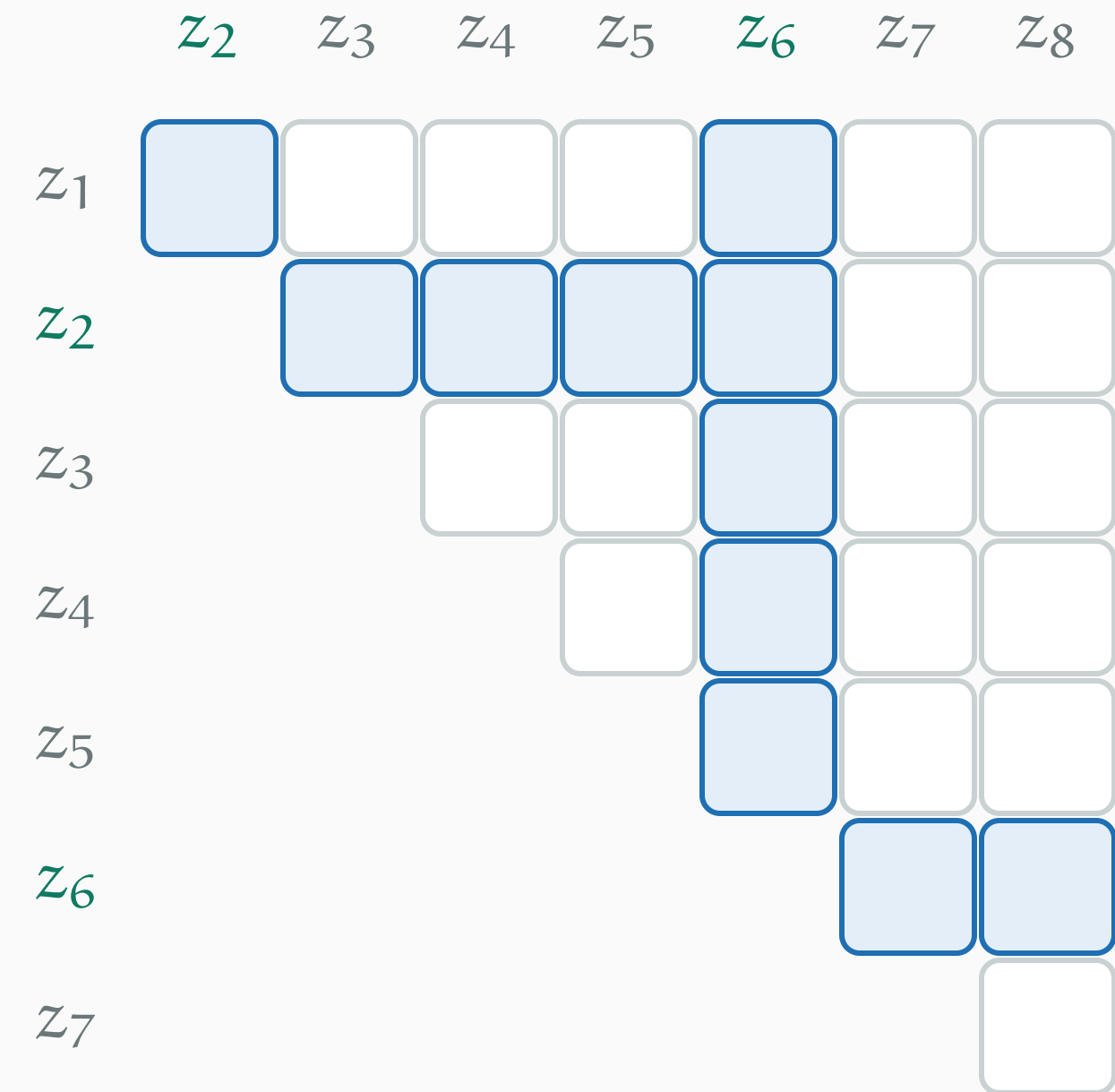
Elements are compared only in PARTITION,
and only with the pivot.

Afterwards the pivot is **in its final place**:
in no later subarray.

So z_i and z_j are compared only when
one of them is the pivot, and then once.

$$X = \sum_{i < j} X_{ij}$$

Every comparison is counted exactly once.

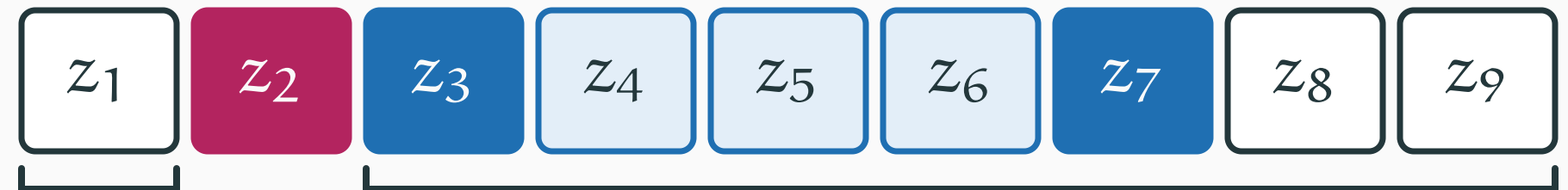


Both calls are over: no pair was compared twice.

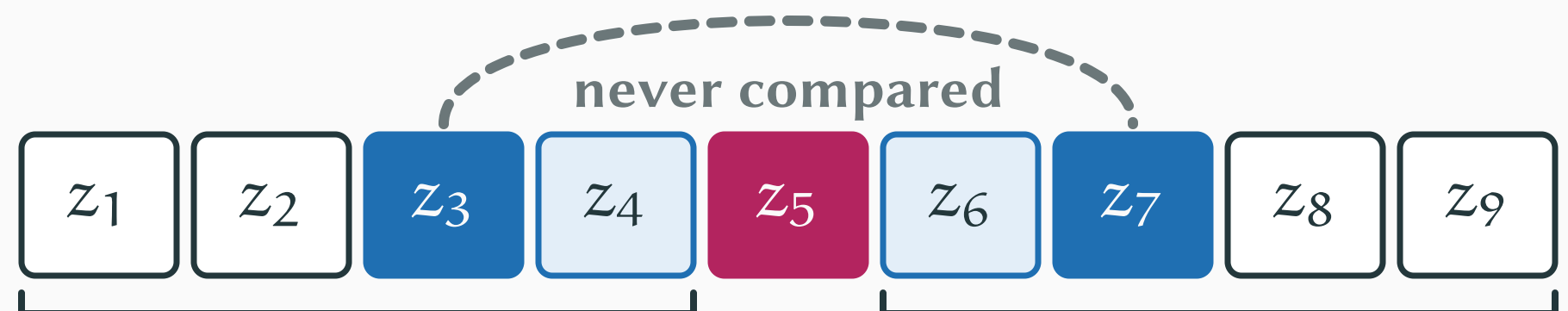
WHEN ARE z_i AND z_j COMPARED?

LEMMA. For $i < j$: z_i and z_j are compared $\iff$ the first pivot chosen from Z_{ij} is z_i or z_j .

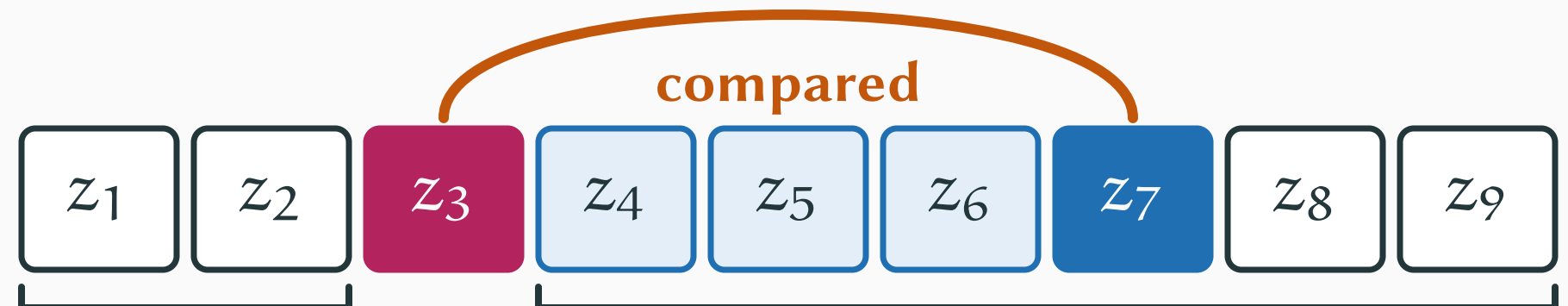
Outside: a pivot outside Z_{ij} (here z_2) sends all of Z_{ij} to the same side.



Between: the first pivot in Z_{ij} is strictly between (here z_5): z_i and z_j are separated for good.



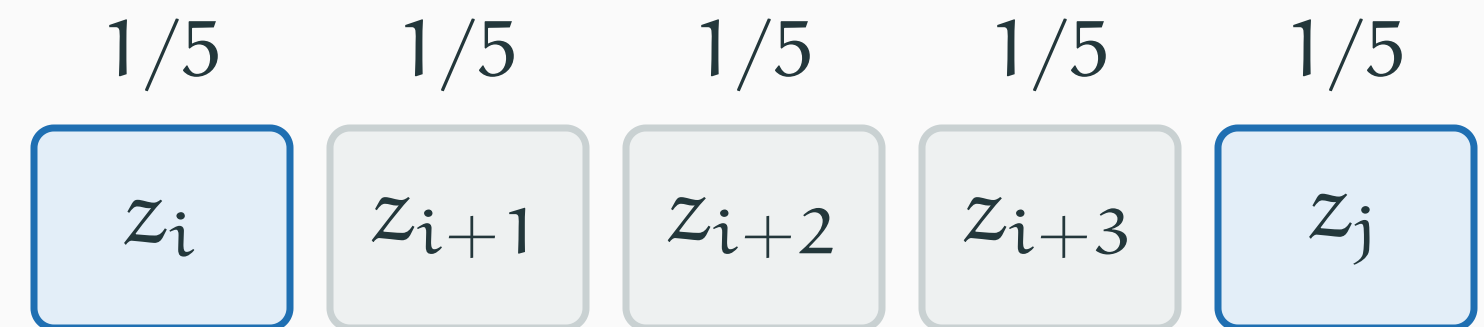
Endpoint: the first pivot is z_i (or z_j): it is compared with the whole subarray, in particular with z_j .



KEY FACT. The first pivot chosen from Z_{ij} is equally likely to be any of its $m = j - i + 1$ elements.

A random pivot does this; so does a random input, since every subarray is again a random ordering.

each: chance of being the first pivot



compared: first pivot is an end

separated: first pivot in the middle

$$\Pr[z_i, z_j \text{ compared}] = \Pr[z_i \text{ is first}] + \Pr[z_j \text{ is first}] \quad (\text{the two events are disjoint})$$

$$= \frac{1}{m} + \frac{1}{m} = \frac{2}{j - i + 1} \quad (\text{key fact})$$

$$\Pr[z_i, z_j \text{ compared}] = \frac{2}{j - i + 1}$$

depends only on the distance $j - i$.

Neighbours: always compared.

The farther apart, the less likely:
 z_1 and z_n with probability $\frac{2}{n}$.

Each row adds up to less than $2 \ln n$;
 all rows together: $\mathbb{E}[X] < 2n \ln n$.

	z_2	z_3	z_4	z_5	z_6	z_7	z_8	row sum
z_1	1	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{2}{5}$	$\frac{2}{6}$	$\frac{2}{7}$	$\frac{2}{8}$	3.44
z_2		1	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{2}{5}$	$\frac{2}{6}$	$\frac{2}{7}$	3.19
z_3			1	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{2}{5}$	$\frac{2}{6}$	2.90
z_4				1	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{2}{5}$	2.57
z_5					1	$\frac{2}{3}$	$\frac{2}{4}$	2.17
z_6						1	$\frac{2}{3}$	1.67
z_7							1	1.00
								16.92

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i < j} X_{ij}\right]$$

(each pair is compared at most once)

$$= \sum_{i < j} \Pr[z_i, z_j \text{ compared}]$$

(linearity; $\mathbb{E}[X_{ij}] = \Pr[X_{ij} = 1]$)

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{2}{j-i+1}$$

(the probability of a comparison)

$$= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k}$$

(put $k = j - i + 1$)

$$< \sum_{i=1}^{n-1} 2H_n < 2nH_n$$

(each inner sum is below $2H_n$)

? *How large is the harmonic number $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$?*

HARMONIC NUMBER.

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}.$$

Upper bound. For $k \geq 2$, the bar of height $\frac{1}{k}$ over $[k-1, k]$ lies under the curve:

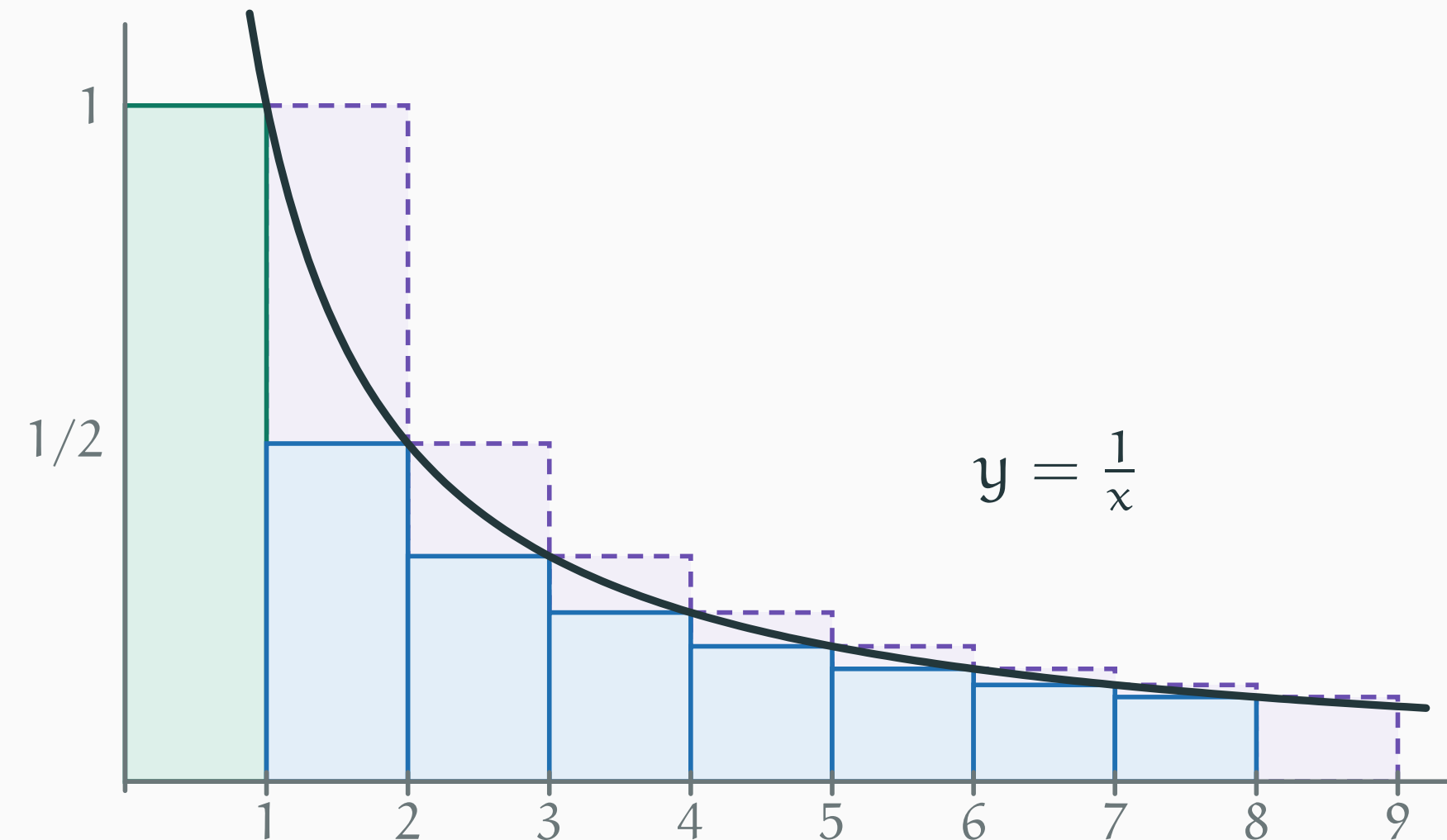
$$H_n - 1 \leq \int_1^n \frac{dx}{x} = \ln n.$$

Lower bound. Bars over $[k, k+1]$ lie above it:

$$H_n \geq \int_1^{n+1} \frac{dx}{x} = \ln(n+1).$$

$$\ln(n+1) \leq H_n \leq 1 + \ln n,$$

so $H_n = \Theta(\log n)$.



$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i < j} X_{ij}\right]$$

(each pair is compared at most once)

$$= \sum_{i < j} \Pr[z_i, z_j \text{ compared}]$$

(linearity; $\mathbb{E}[X_{ij}] = \Pr[X_{ij} = 1]$)

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{2}{j-i+1}$$

(the probability of a comparison)

$$= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k}$$

(put $k = j - i + 1$)

$$< \sum_{i=1}^{n-1} 2H_n < 2nH_n$$

(each inner sum is below $2H_n$)

$$\leq 2n(1 + \ln n) = O(n \log n)$$

($H_n \leq 1 + \ln n$)

Group the pairs by distance $d = j - i$:
 $n - d$ pairs, each with probability $\frac{2}{d+1}$.

$$\mathbb{E}[X] = \sum_{d=1}^{n-1} (n - d) \frac{2}{d+1}.$$

Put $m = d + 1$, so $(n - d) \frac{2}{d+1} = \frac{2(n+1)}{m} - 2$:

$$\begin{aligned} \mathbb{E}[X] &= \sum_{m=2}^n \left(\frac{2(n+1)}{m} - 2 \right) \\ &= 2(n+1)(H_n - 1) - 2(n-1). \end{aligned}$$

$$\mathbb{E}[X] = 2(n+1)H_n - 4n$$

THEOREM. On a uniformly random input, Quicksort runs in expected time $O(n \log n)$.
For $n \geq 2$, the expected number of comparisons is

$$\mathbb{E}[X] = 2(n+1)H_n - 4n < 2n \ln n \approx 1.39 n \log_2 n.$$

Random pivot instead of random input:

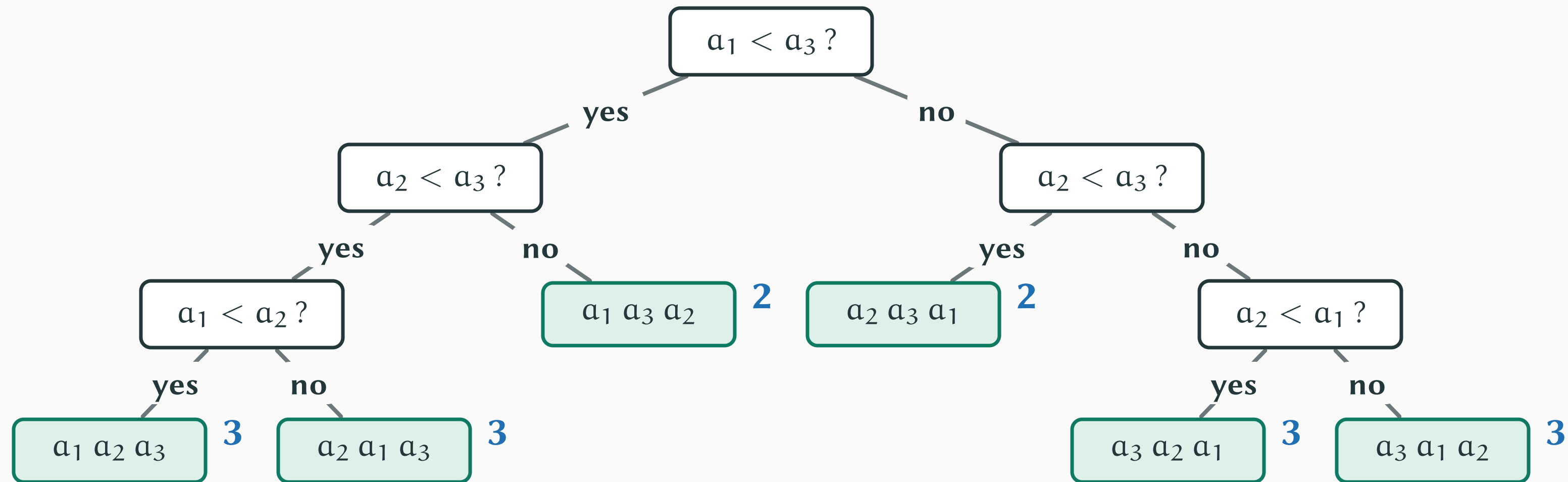
the same bound for *every* input.

No bad inputs, only bad luck.

? *Can a version of Quicksort have worst-case time $O(n \log n)$?*

Yes: take the **median** as pivot, found in $O(n)$ time
by the selection algorithm: $T(n) = 2T(n/2) + O(n)$.

Quicksort on a_1, a_2, a_3 , run on all $3! = 6$ orderings:
each node is a comparison, each leaf an output.



Sum of the depths: $3 + 3 + 2 + 2 + 3 + 3 = 16$. Average over the 6 inputs: $16/6 = 8/3 \approx 2.67$.

Every comparison sort has such a **decision tree**.

The worst case is the **depth** d of the tree:
the longest path from the root to a leaf.

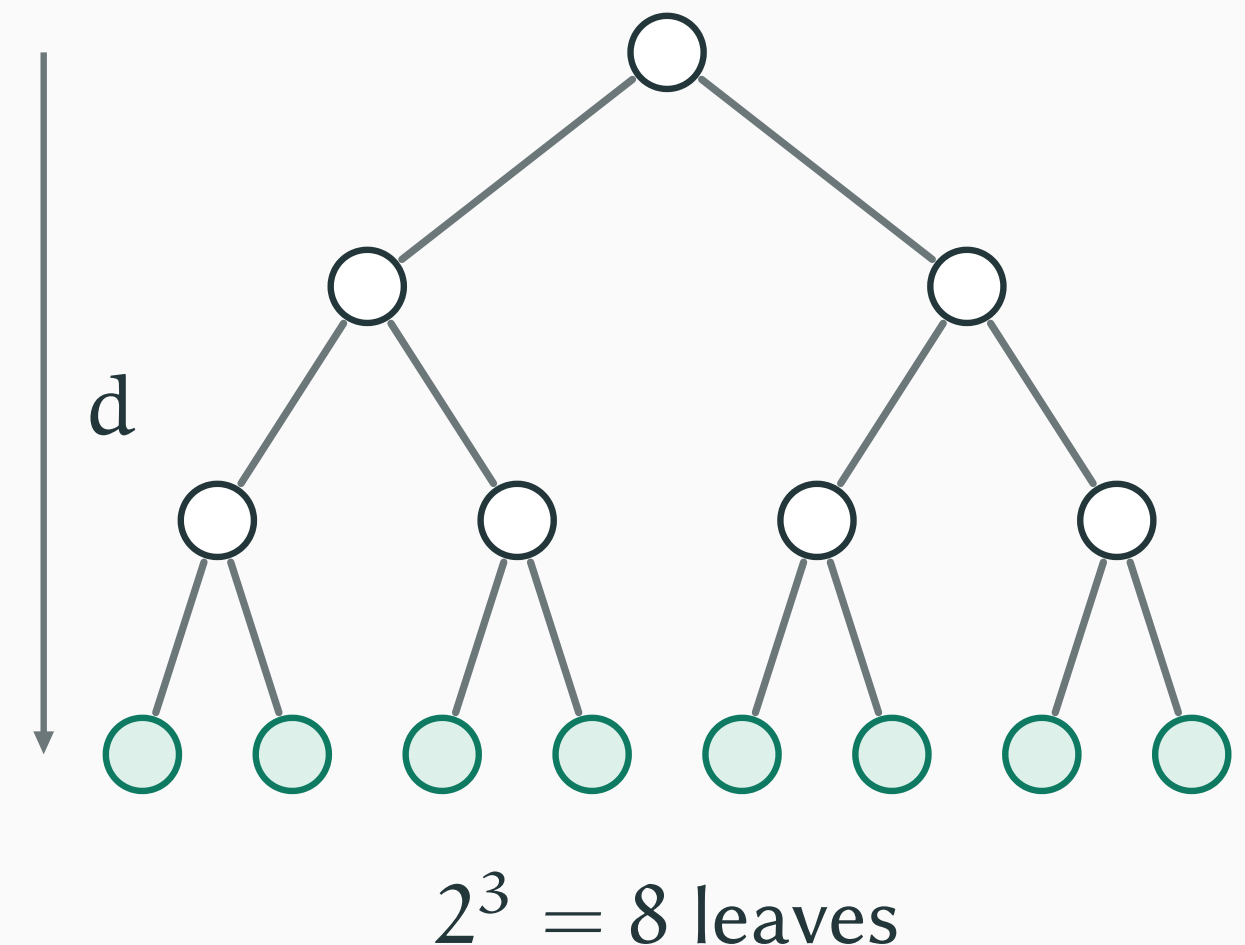
At least $n!$ leaves, one per output.

A binary tree of depth d has at most 2^d leaves.

So $2^d \geq n!$, that is $d \geq \log_2(n!)$.

$n! = 1 \cdot 2 \cdots n \geq \left(\frac{n}{2}\right)^{n/2}$ (at least $n/2$ factors are $\geq n/2$)

$$d \geq \log_2(n!) \geq \frac{n}{2} \log_2 \frac{n}{2} = \Omega(n \log n)$$

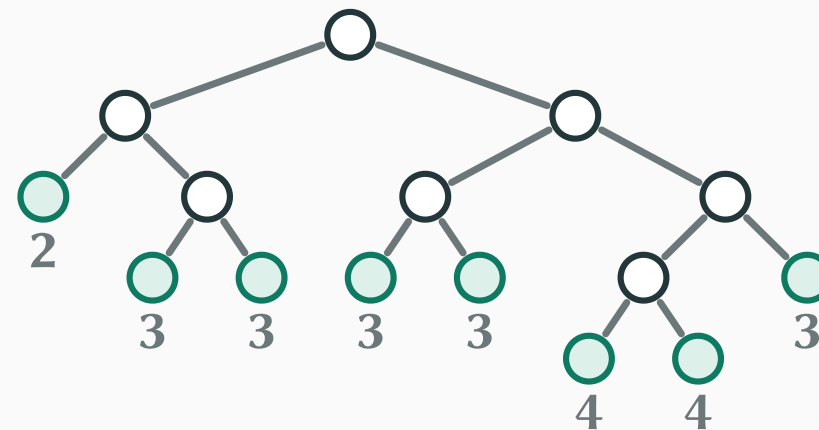


On average over the $n!$ inputs, the cost is the **average depth of a leaf**.

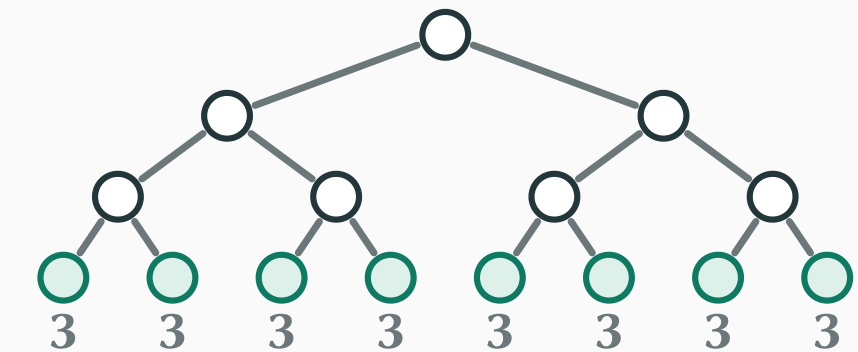
? Which binary tree with 8 leaves has the smallest average depth?



sum of depths 35
average $35/8 = 4.38$



sum of depths 25
average $25/8 = 3.13$



sum of depths 24
average $24/8 = 3.00$

The balanced tree: $\log_2 8 = 3$. Can a tree with L leaves beat $\log_2 L$?

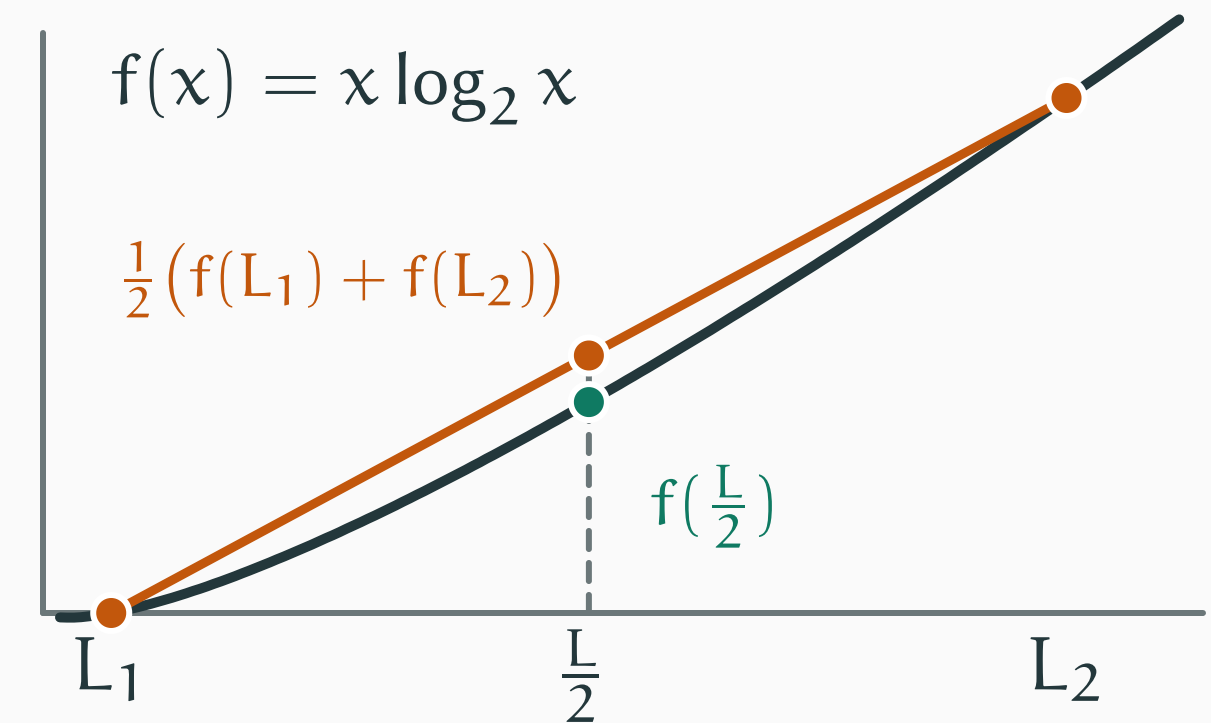
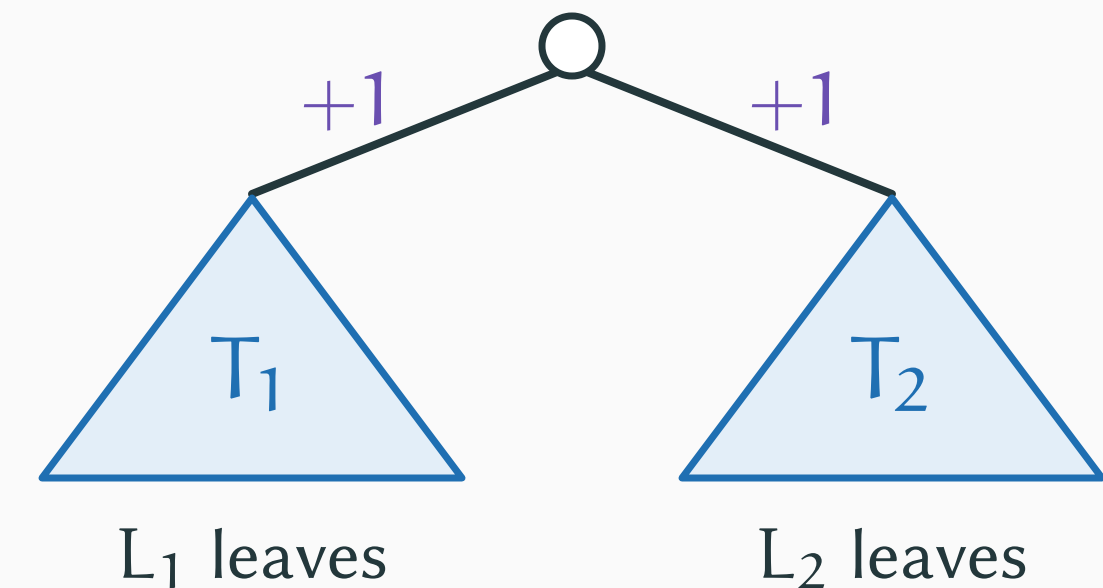
LEMMA. Let T be a binary tree with L leaves in which every internal node has two children, and $D(T)$ the sum of the depths of its leaves. Then $D(T) \geq L \log_2 L$.

Induction on L . $L = 1$: $D = 0 = 1 \cdot \log_2 1$.

$L \geq 2$: the root has subtrees T_1, T_2 with $L_1 + L_2 = L$ leaves.

$$\begin{aligned}
 D(T) &= D(T_1) + D(T_2) + L && \text{(each leaf is one level deeper in } T) \\
 &\geq f(L_1) + f(L_2) + L && \text{(by induction; } f(x) = x \log_2 x) \\
 &\geq 2f\left(\frac{L}{2}\right) + L && \text{(} f \text{ is convex)} \\
 &= L \log_2 L. \quad \square && (2f\left(\frac{L}{2}\right) = L \log_2 L - L)
 \end{aligned}$$

So the average depth of a leaf is $D(T)/L \geq \log_2 L$.



THEOREM. Every comparison sort makes at least $\log_2(n!) = \Omega(n \log n)$ comparisons on average, over the $n!$ orderings of the input.

The $n!$ inputs need $n!$ different outputs, so they end in $n!$ different leaves.

Average number of comparisons
= average depth of these leaves
 $\geq \log_2(n!)$ (by the lemma).

And $\log_2(n!) \geq \frac{n}{2} \log_2 \frac{n}{2} = \Omega(n \log n)$.

n	$\log_2(n!)$	Quicksort	ratio
5	6.9	7.4	1.07
10	21.8	24.4	1.12
100	524.8	647.9	1.23
1000	8529.4	10985.9	1.29

Quicksort's average against the bound: factor $\rightarrow 2 \ln 2 \approx 1.39$.

Comparisons to sort n distinct numbers (average: over all $n!$ orderings).

	average	worst case
Merge sort	$\approx n \log_2 n$	$\approx n \log_2 n$
Quicksort	$\approx 1.39 n \log_2 n$	$\approx n^2/2$
Every comparison sort	$\geq \log_2(n!) \approx n \log_2 n$	$\geq \log_2(n!)$

Quicksort makes about 39% more comparisons than merge sort, yet is often faster in practice:
it sorts in place, with a simple inner loop.

- 1 **Average case needs a model:**
a uniformly random input (or a random pivot).
Tools: expectation, indicators, linearity.
- 2 **Quicksort:** z_i, z_j are compared iff the first pivot from Z_{ij} is one of them: probability $\frac{2}{j-i+1}$.
So $\mathbb{E}[X] = 2(n+1)H_n - 4n = O(n \log n)$.
- 3 **Lower bound:** a comparison sort is a decision tree with $n!$ leaves; its average leaf depth is at least $\log_2(n!) = \Omega(n \log n)$.

Quicksort is optimal on average,
up to a factor of about 1.39.

Reference: CLRS, Chapters 7.1, 7.3, 7.4