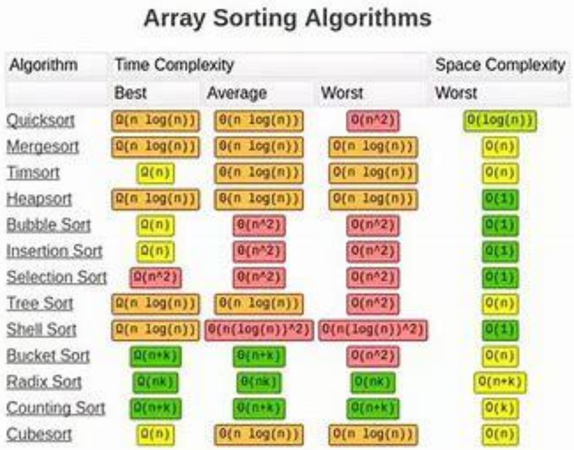


Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

Design and Analysis of Algorithms I

Merge Sort (Analysis)

RUNNING TIME OF MERGE SORT

Claim: For every input array of n numbers, Merge Sort produces a sorted output array and uses at most $6n\log_2 n + 6n$ operations.

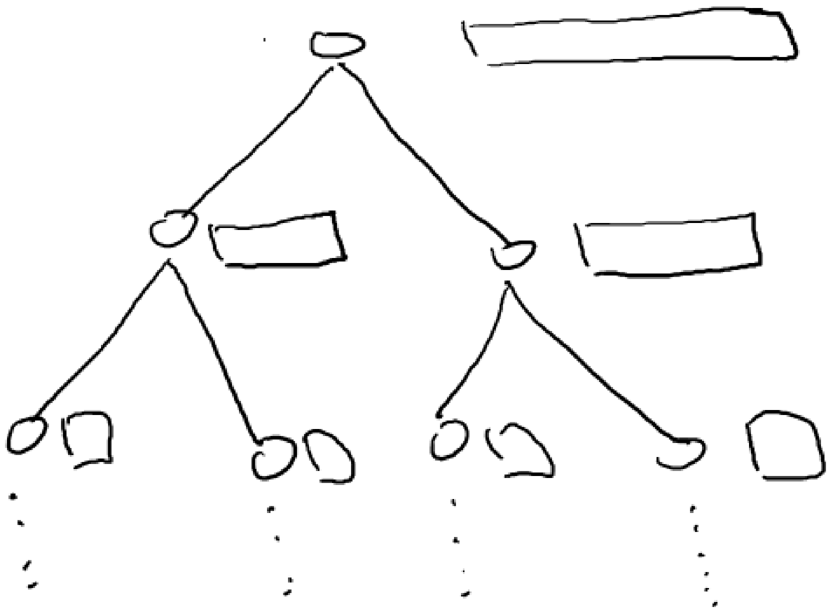
PROOF OF CLAIM (ASSUMING N = POWER OF 2):

root

Level 0 [outer call to Merge Sort]

Level 1
(1st recursive calls)

Level 2



Nextcore AI -
Gopal
Shangari

Te
ve

Roughly how many levels does this recursion tree have (as a function of n, the length of the input array)?

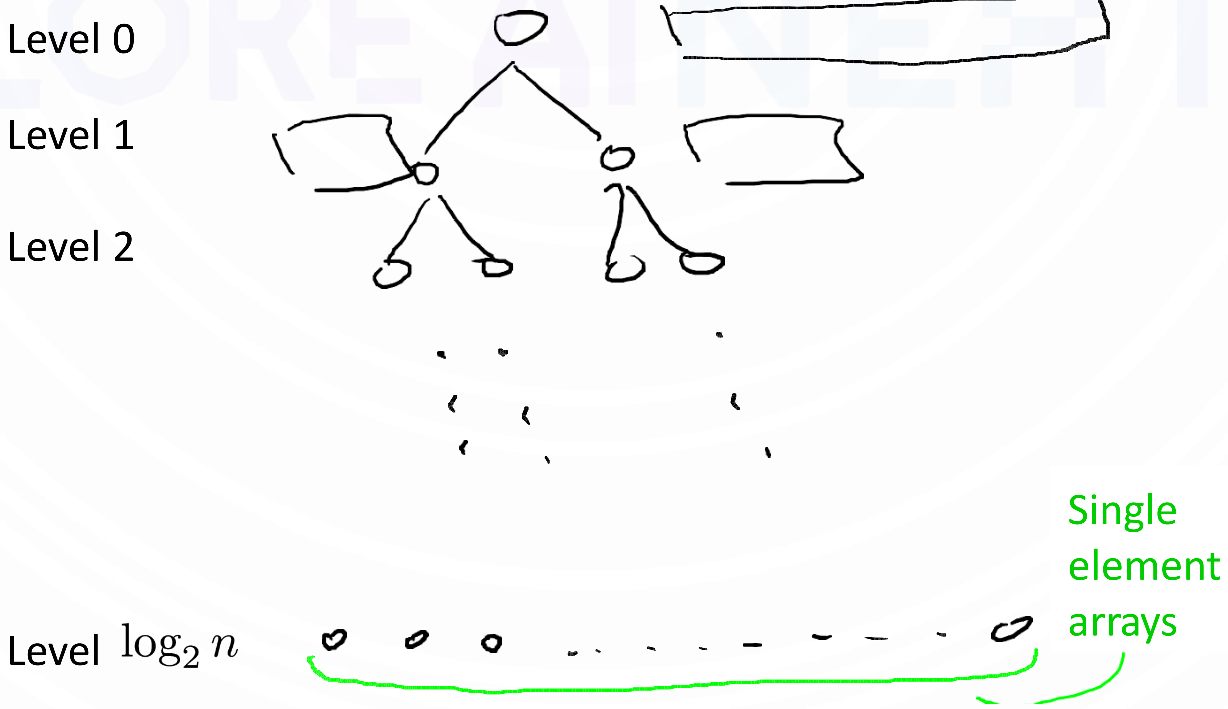
○ A constant number (independent of n)

○ $\log_2 n$ $(\log_2 n + 1)$ to be exact!

○ $\sqrt{n}$

○ n

PROOF OF CLAIM (ASSUMING N = POWER OF 2):



Level 0

Level 1

Level 2

Level $\log_2 n$

Single element arrays

What is the pattern ? Fill in the blanks in the following statement: at each level j = 0,1,2,.., $\log_2 n$, there are <blank> subproblems, each of size <blank>.

○ 2^j *and* 2^j, *respectively*

○ $n/2^j$ *and* $n/2^j$, *respectively*

[✓ ○ 2^j *and* $n/2^j$, *respectively*]

○ $n/2^j$ *and* 2^j, *respectively*

PROOF OF CLAIM (ASSUMING N = POWER OF 2) :

At each level j=0,1,2,.., $\log_2 n$,

Total # of operations at level j = 0,1,2,..., $\log_2 n$

$$\le 2^j * 6(\frac{n}{2^j}) = 6n$$

# of level-j subproblems

Size of level-j subproblem

Work per level – j subproblem

$$\boxed{\text{Total}}$$

$$6n(\log_2 n + 1)$$

Work per level

# of levels

RUNNING TIME OF MERGE SORT

Claim: For every input array of n numbers, Merge Sort produces a sorted output array and uses at most $6n\log_2 n + 6n$ operations.

QED!