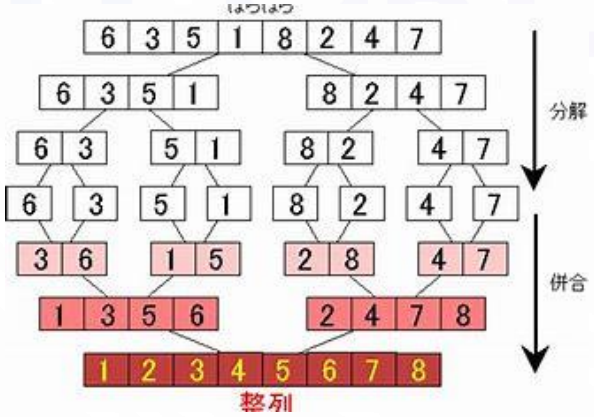
NEXTCORE AI

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

Introduction

Merge Sort (Pseudocode)

MERGE SORT: PSEUDOCODE

- recursively sort 1st half of the input array
- recursively sort 2nd half of the input array
- merge two sorted sublists into one

  [ignores base cases]

PSEUDOCODE FOR MERGE:

C = output [length = n] A
= 1st sorted array [n/2] B
= 2nd sorted array [n/2]
i= 1
j = 1

```
for k = 1 to n
        if A(i) < B(j)
            C(k)= A(i)
            i++

        else [B(j) < A(it)
            C(k) = B(j)
            j++
end
```

(ignores end cases)

Nextcore AI -
Gopal Shangari

MERGE SORT RUNNING TIME?

Key Question : running time of Merge Sort on array of n numbers ?

[running time ~ # of lines of code executed]

PSEUDOCODE FOR MERGE:

C = output [length = n] A
= 1st sorted array [n/2] B
= 2nd sorted array [n/2]
i = 1
j = 1

} 2 operations

```
for k = 1 to n
    if A(i) < B(j)
        C(k)= A(i)
        i++

    else [B(j) < A(it)
        C(k) = B(j)
        j++

end
```

(ignores end cases)

Nextcore AI -
Gopal Shangari

RUNNING TIME OF MERGE

Upshot : running time of Merge on array of m numbers is $\leq 4m + 2$

$\leq 6m$ (Since $m \geq 1$)

RUNNING TIME OF MERGE SORT

Claim : Merge Sort requires $\leq 6n \log_2 n + 6n$ operations to sort n numbers.

Recall : $= \log_2 n$ is the # of times you divide by 2 until you get down to 1