Recursion 2

18th Jan - Wednesday Optional loubt ression

25th Jan - Wednerday

Agenda

- Complexity Analysis
- Merge two sorted arrays
- Mergesort

Dry Run - sum(N) 1) Court apprex us of recursive calls for input N. 3 (N toi) muz 2) find out cost of each cell return sum (N-1) 0(N) 3 sum(s) { - ret sum(y) + 510+5 __ ret sum (3) + 4 3 - vet <u>sum(2)</u> + 3 sum (2) { ret rum (1) + 2 Input 5: 3 Recursive Cally: 5 Imput: N Recursive Calls: N Cost of each recurrine coll - O(1) Total = N × O(1) = O(N)

Dry Run - Factorial

fact (int N) {

$$V = 3$$
 $V = 3$
 $V =$

return fact(N-1) x N

 $\begin{cases}
fact(3) & \xi \\
 & || N=3
\end{cases}$ ret $fact(2) \times 3$ $\begin{cases}
3
\end{cases}$ $\begin{cases}
fact(2) \times 3
\end{cases}$

fact (2) {
// N=2

// N=2

// Add (1) X 2

// Jack (1) Jack (1)

Input: I Recurrive Cells: 4

Input: N Recurive Calls: NH

Cost of : O(1) each frame

Lact (0) E 11 N=0 Rase Case 3

Total Time Complaity = (N+1) x O(1) = O(N)

Dry Run - Fibonacci

fib (int N)
$$\xi$$

if (N==0 or N==1)

veturn 1

return fib(N=1) + fib(N=2)

Recurrion Tree

No of calls

$$1 = 2^{n}$$
 $1 = 2^{n}$
 $1 = 2^{n}$

Total recurive calls =

1 + 2 + 4 + 8 + 16 + 2^{N-1} = 2^{N} -1

Carl series

Total time

Complexity = $3(2^{N})$

Merge two sorted arrays



Given two sorted arrays, generate a new sorted array from them in O(n) time complexity.

Example 1

$$A = -3$$
, 10, 14, 18, 28, 31, 35

 $B = -5$, 0, 4, 6, 12, 18, 30

 $A = -5$, 0, 4, 6, 10, 12, res. append (ACi3)

 $A = -5$, 28, 30, 31, 35

else

 $A = -5$, 0, 5, 10, 20, 70, 40, 50

 $A = -5$, 0, 5, 10, 20, 70, 40, 50

 $A = -5$, 0, 1, 2, 3, 5, 10, 20, 70, 40, 50

 $A = -5$, 0, 1, 2, 3, 5, 10, 20, 70, 40, 50

Mergesort

In merge sort, the given array is divided into roughly two equal subarrays. These sub-arrays are divided over and over again into halves until we end up with arrays having only one element each. At last, we merge the sorted pairs of sub-arrays into a final sorted array.



Divide this array juts roughly equal part

Time Complexity = O(N log N)
Now? - Try it on your own

WINT

 $N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \frac{N}{8}$

Doubts

Thank

Wednesday - Optional Doubt Session

Archive Classes section - support @ scaler.com

Name algorithms

-> Bubble
-> Binary
-> Invertion
-> Selection

n Merge

Aft Call

Uber > Cro

Tava

Response

Tail Call Recursion

Ly Optimize recursive calls

Advanced Recursion

Clood Night You

Wednesday