Divide & conquer

n is large ⟶ Big → Divide & conquer

n is small ⟶ small → return solution

↳ size of elements

Recursion → Calling the function itself directly or indirectly

↓

Recursive tree ⟶ Recurrence Relation

↳ 1) substitution
2) Recursive tree
3) Master's Theorem

⟵ Time Complexity

Pseudocode

arr → array, p → lower index (0), q → higher index (len(arr)-1)

T(n) ⟶ divideAndConquer(arr, p, q):

c ⟶ { if small(arr, p, q):
return solution

else:

Divide — ① ⌜

c ⟶ mid = Divide(arr, p, q) ⌟ T(n/2)

conquer — { b = divideAndConquer(arr, p, mid)
c = divideAndConquer(arr, mid+1, q)

↳ Recursion

T(n/2) T(n/2)

conquer ② ⌟

return combine(b, c) ⟶ c

↳ combine

↳ ③

↳ optional

Recurrence Relation

$$T(n) = 2T(n/2) + c$$

$$T(n) = aT(n/b) + f(n)$$

Binary Search

$$T(n) = T(n/2) + c$$

QuickSort

$$T(n) = 2T(n/2) + n$$

find max. element in an array

↳ small problem