Divide and conquer

Divide and conquer

- Algorithmic technique of splitting problem into smaller instances of the same problem, solving them recursively, and combining the results

- Examples: mergesort, quicksort, binary search

Finding running time: Recurrences

Consider running time for mergesort

Let T(n) be the time to sort n elements

Finding running time: Recurrences

Consider running time for mergesort

Let T(n) be the time to sort n elements

The algorithm requires

- Calls to sort n/2 elements — T(n/2) each
- The merge (and other stuff) — O(n)

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

Finding running time: Recurrences

Consider running time for mergesort

Let T(n) be the time to sort n elements

The algorithm requires

- Calls to sort n/2 elements — T(n/2) each
- The merge (and other stuff) — O(n)

$$T(n) = 2T(n/2) + \cancel{O(n)}\ n$$

Caveat: Imperfect division

- The subproblem size isn't exactly n/2
  - should involve floors and ceilings
  - other problems might be off by additive constant

    e.g. $T(n) = 2T(n/2 + 3) + n$

- Turns out not to matter for asymptotic answer

What is the recurrence for binary search?

A. $T(n) = T(n/2) + 1$
B. $T(n) = 2T(n/2) + 1$
C. $T(n) = T(n/2) + n$
D. $T(n) = 2T(n/2) + n$
E. None of the above

What is the recurrence for binary search?

A. $T(n) = T(n/2) + 1$
B. $T(n) = 2T(n/2) + 1$
C. $T(n) = T(n/2) + n$
D. $T(n) = 2T(n/2) + n$
E. None of the above