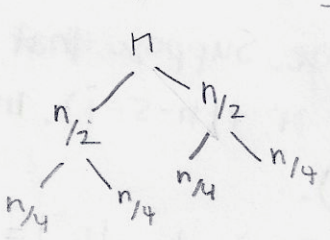
CS157 Lecture #7 : DIVIDE & CONQUER

① split into two problems, then merge the results

$f(n) = 2f(\frac{n}{2}) + n$ ← splitting & combining



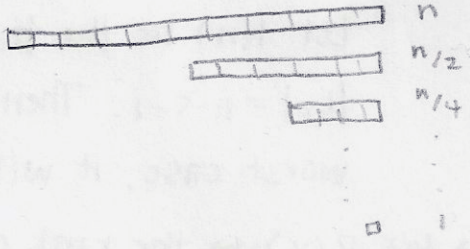
n work at each level

log n levels

$O(n \log n)$

② split into two problems, but solve only 1 (the splitting tells you which to solve)

$f(n) = f(\frac{n}{2}) + n$



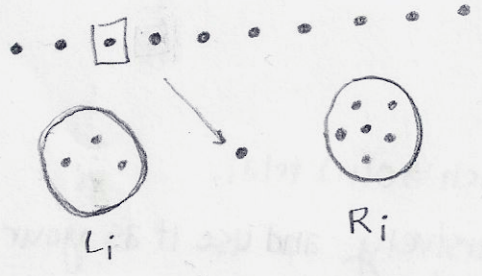
$O(n)$

FINDING MEDIANS → How long does it take? (depends on algorithm)

↳ aiming for linear time

↘ SORTING: $O(n \log n)$

IDEA



list of size n
choose a pivot element
sort elements into two "piles": smaller than pivot / larger than pivot

recur on pile that contains the $\lceil n/2 \rceil$th smallest elem.

Goal for recursion : Given a list of size n, select the ith smallest in $O(n)$

ALGO ↳ select (L, i): ← distinct elem

x ← first elem of L
L_1 ← all elem of L that are < x
L_2 ← all elem of L that are > x
if $|L_1| \geq i$, return select(L_1, i)
else if $|L_1| = i-1$, return x
else return select(L_2, $i-(|L_1|+1)$)

RUNTIME : depends on how "lucky" we are

↓

"unlucky" if x is always the largest or smallest elem of L (e.g. L is sorted)

↳ one iteration takes time $|L|$

↳ select(sorted list, n) = $n + n-1 + n-2 + \cdots = O(n^2)$!!

* instead of choosing x to be the first element, let it be a random element

* "Security by Obscurity": protect yourself from malicious inputs by obscuring what algorithm you're using (e.g. don't let them know how you're choosing your pivot)

* Assume fixed input (since you can't control that). Prove that average case (i.e. average over choice of pivot) runs in $O(n)$ time.

  ↳ Claim: runtime depends only on the size of L, not L itself (or i)

  Let $t(n)$ be the time the algo takes on average. Suppose that $|L_1| = 5$, $|L_2| = n-5-1$. Then we either need to do $t(5)$ or $t(n-5-1)$. In the worst case, it will be $\max(t(5), t(n-5-1))$.

  ↳ let $R(x)$ be the rank of x in L. Then $|L_1| = R(x)-1$, $|L_2| = n-R(x)$ then the time will be at most $\max(t(|L_1|), t(|L_2|))$. When $|L_1| = |L_2|$, then $t(n) = t(n/2) + n$ ⇒ we're in good shape. (REALLY goodshape)

  If $\frac{n}{4} \le R(x) \le \frac{3n}{4}$, then $|L_1| \le \frac{3n}{4}$, $|L_2| \le \frac{3n}{4}$ ⇒ $t(n) \le t(\frac{3n}{4}) + n$.

  The probability of this happening is $\frac{1}{2}$, since the distribution of $R(x)$ is uniform. ⇒ Takes a constant # of "coin flips" (choices of random #) to get to this case ⇒ takes time $O(n)$ [expected to take this amt of time]

★ How can we select a pivot **deterministically**?

① divide the list into groups of 5 elements

② for each group, find its median → $O(1)$ for each ⇒ $O(n)$ total

③ find the median of these $\frac{n}{5}$ elements recursively and use it as your pivot

↳ Why does this guarantee a good pivot? If there are n elements total, then the pivot is greater than at least $\frac{3n}{10}$ elem, and less than at least $\frac{3n}{10}$ elem ⇒ $\frac{3n}{10} \le R(\text{pivot}) \le \frac{7n}{10}$ ⇒ $t(n) \le \underbrace{t(\frac{7n}{10})}_{\text{recursion after choosing x}} + \underbrace{t(\frac{n}{5})}_{\text{calculate x}} + \underbrace{n}_{\text{work @ each level}}$, base case 1.

Claim: $t(n) \le 1000n$

Pf: By the inductive hypothesis, $t(\frac{7n}{10}) + t(\frac{n}{5}) + n \le 700n + 200n + n = 901n$

But $t(n) \le t(\frac{7n}{10}) + t(\frac{n}{5}) + n \le 901n \le 1000n$ ∎