CS 3510

Goals:

Thinking abstractly

Thinking recursively

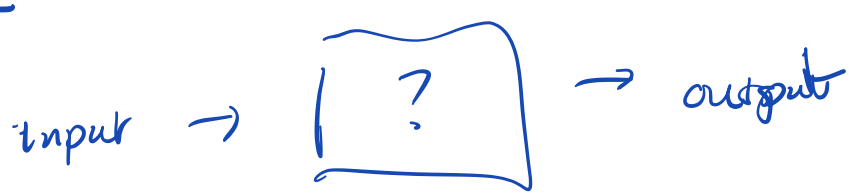
Design + analysis of algorithms

Binary search

DFS

sorting

addition



correct?

How fast?

# of operations: { # of comparisons; # bit / digit operations; # data structure operations

How fast

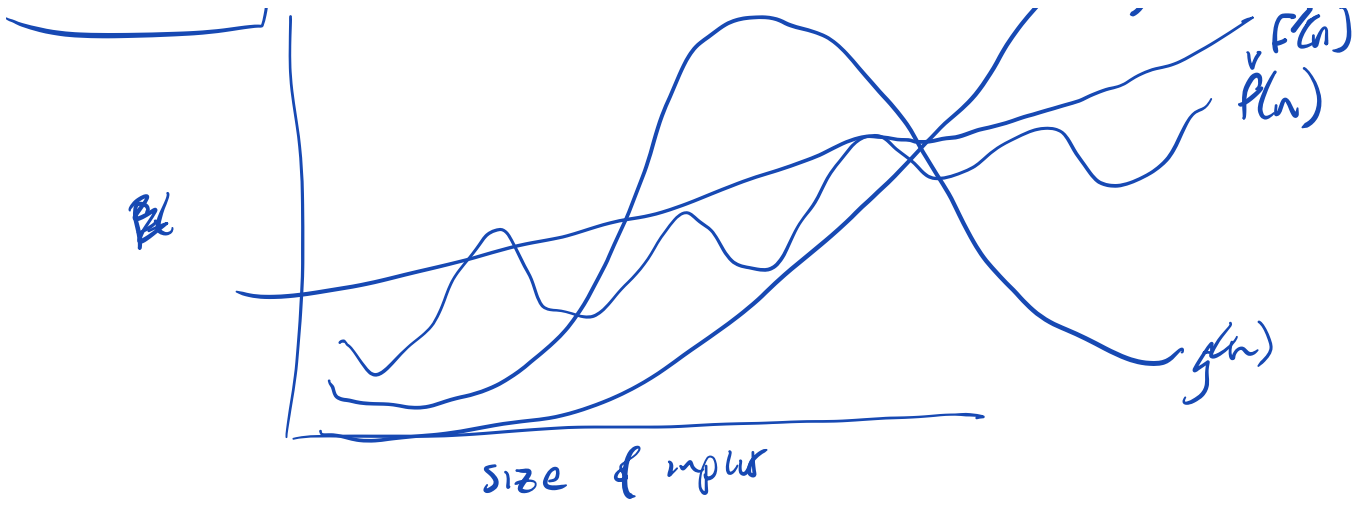
operations
comparisons
data str operations

input → f(n) steps → output

How fast

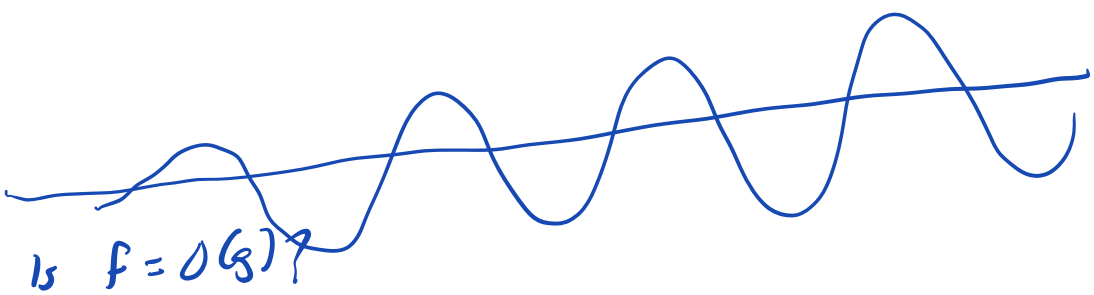
does f(n) grow?

h(n)



Def: $f(n)$ is $O(g(n))$ if

$\exists\ c > 0,\ n_0$ s.t. $f(n) \le c \cdot g(n)$ $\forall\ n \ge n_0$



Is $f = O(g)$?

Will $c \cdot g(n)$ overtake f at some point & from then on?

I Divide & conquer

II Dynamic programming

III Graph algorithms

IV NP-completeness

V optional topics

I Divide + conquer.

A) Divide the problem into smaller subproblems (of the same type)

B) Solve subproblems recursively

C) Merge the solutions to solve the original problem.

Merge sort

Given $a_1, \ldots, a_n$ ← distinct and we want to output them in order.

Operation: # comparisons

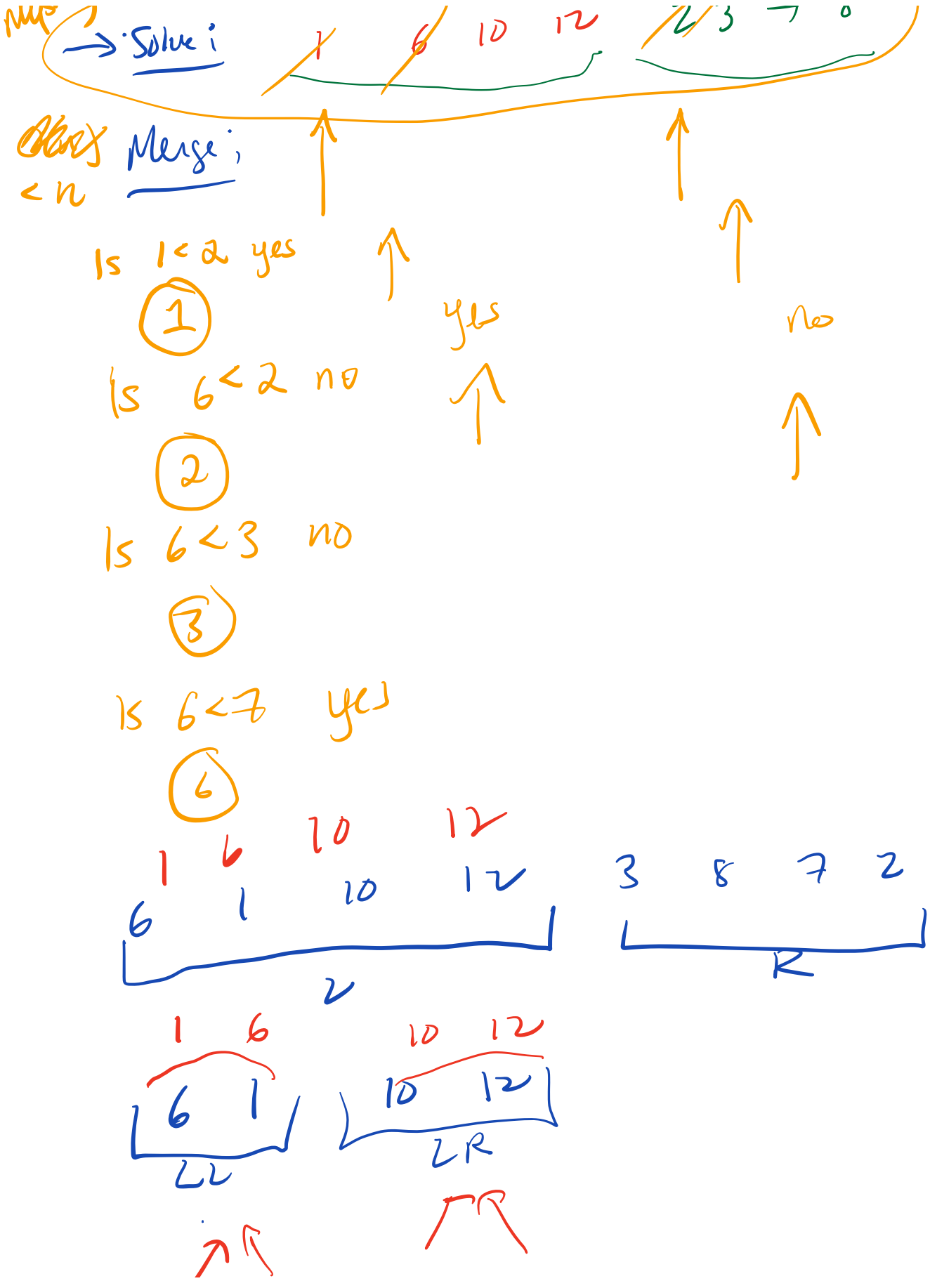
Is $a_i < a_j$?

Assume $n = 2^k$ for some $k \in \mathbb{Z}^+$

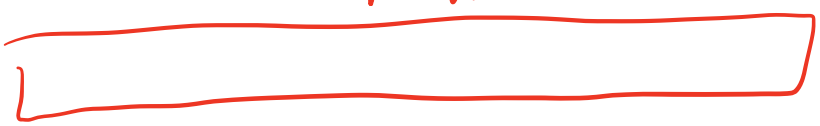
Ex:

sort 6 1 10 12 3 8 7 2

Divide: L R


Solve i
Merge;
< n
Is 1<2 yes
1
Is 6<2 no
2
Is 6<3 no
3
Is 6<7 yes
6
yes
no
1 6 10 12
6 1 10 12
3 8 7 2
L
R
1 6
6 1
LL
10 12
10 12
LR

16 11 10 12



Time to sort n items is $T(n)$

$$T(n) \leq 2T\left(\frac{n}{2}\right) + n$$

$$T(n) \leq 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n$$

$$= 4T\left(\frac{n}{4}\right) + 2\cdot\frac{n}{2} + n$$

$$= 4T\left(\frac{n}{4}\right) + 2n$$

$$T(n) = 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n$$

$$= 8T\left(\frac{n}{8}\right) + \frac{4n}{4} + 2n$$

$$= 8T\left(\frac{n}{8}\right) + 3n$$

$$T(n) \geq 16\,T\left(\frac{n}{16}\right) + 4n$$

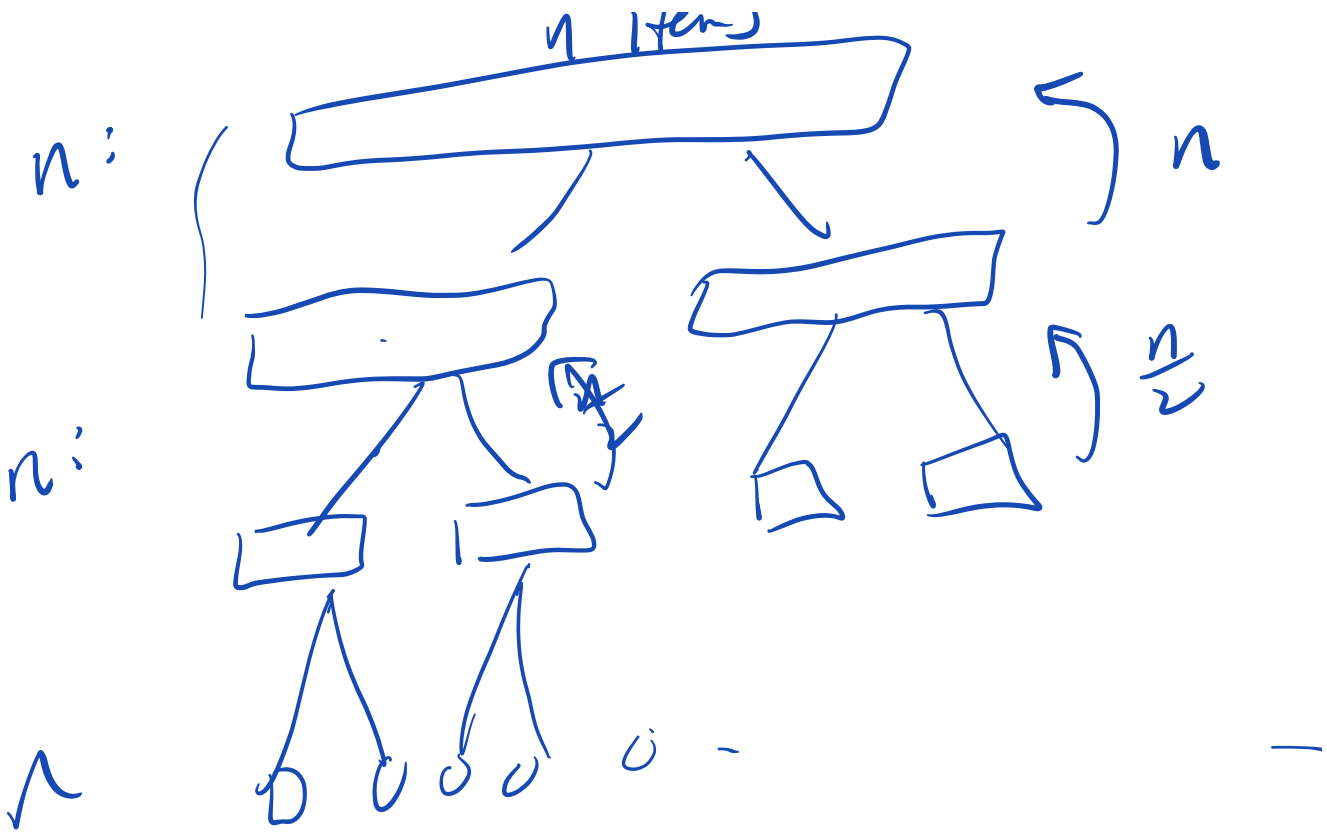
Guess

$$T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn$$

If $T(n) \leq 2^k T\left(\frac{n}{2^k}\right) + kn$

then $T(n) \leq 2^{k+1} T\left(\frac{n}{2^{k+1}}\right) + (k+1)n$

$T(1) = 0$



Total # comp $\leq k \cdot n$

$= \log_2 n \cdot n$