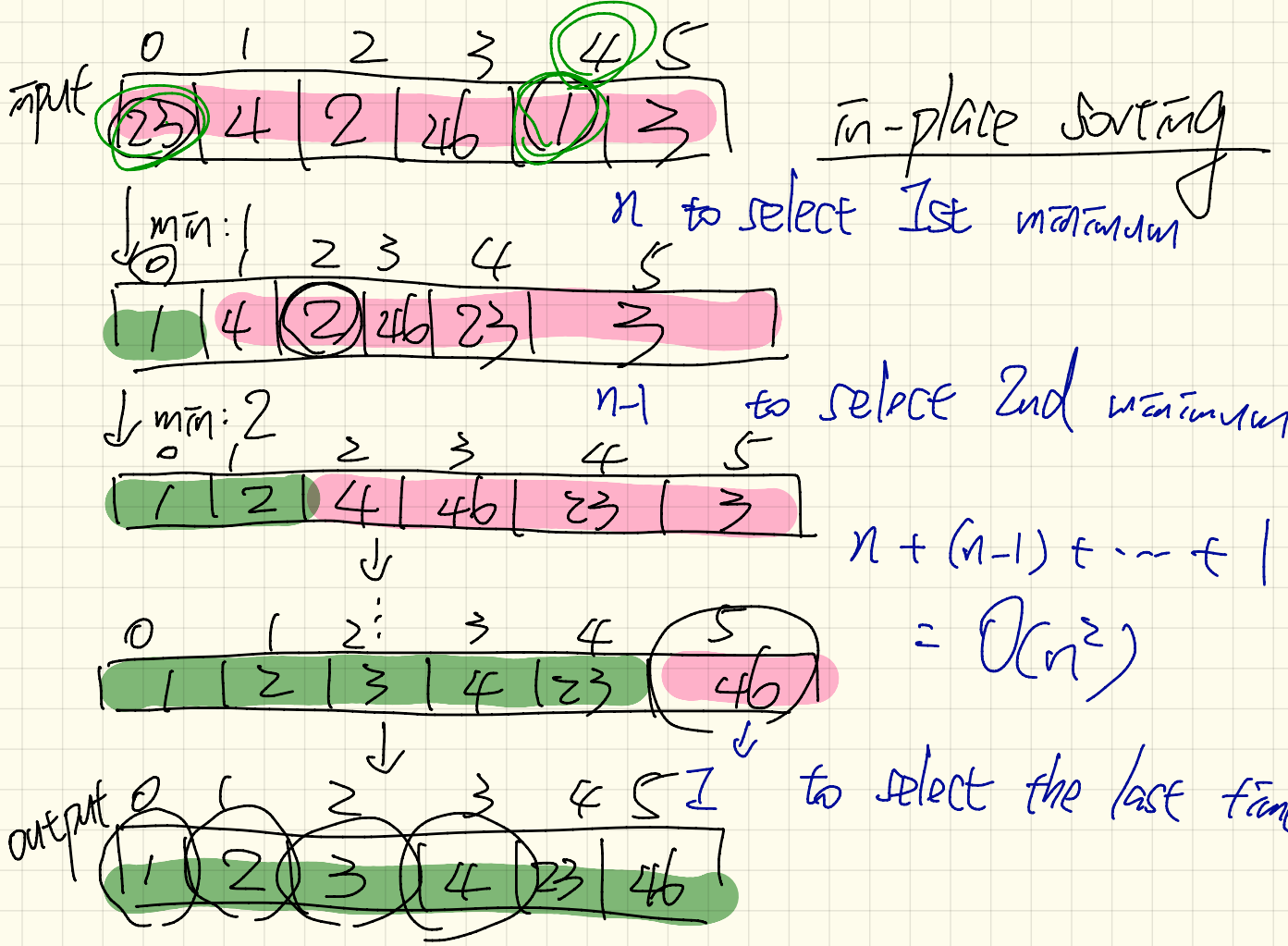
Lecture 18

Thursday Nov. 9


0 1 2 3 4 5
input
23 4 2 46 1 3
in-place sorting
n to select 1st minimum
min: 1
0 1 2 3 4 5
1 4 2 46 23 3
min: 2
n-1 to select 2nd minimum
0 1 2 3 4 5
1 2 4 46 23 3
n + (n-1) + ... + 1
= O(n²)
0 1 2 3 4 5
1 2 3 4 23 46
1 to select the last times
output
0 1 2 3 4 5
1 2 3 4 23 46

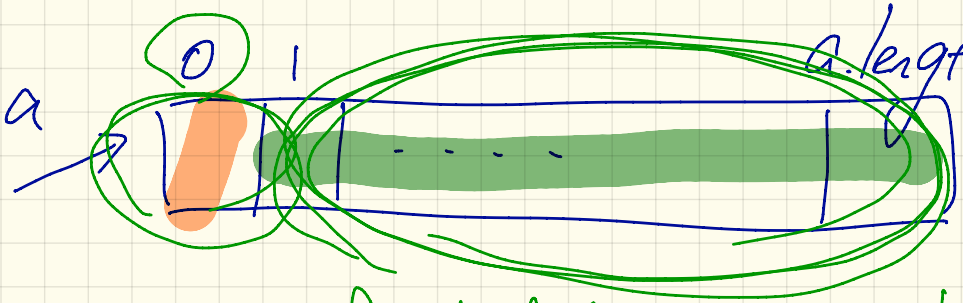
Problem: Recursively find min from an array

(assume a.length >= 1)

```
int findMin(int[] a)
```

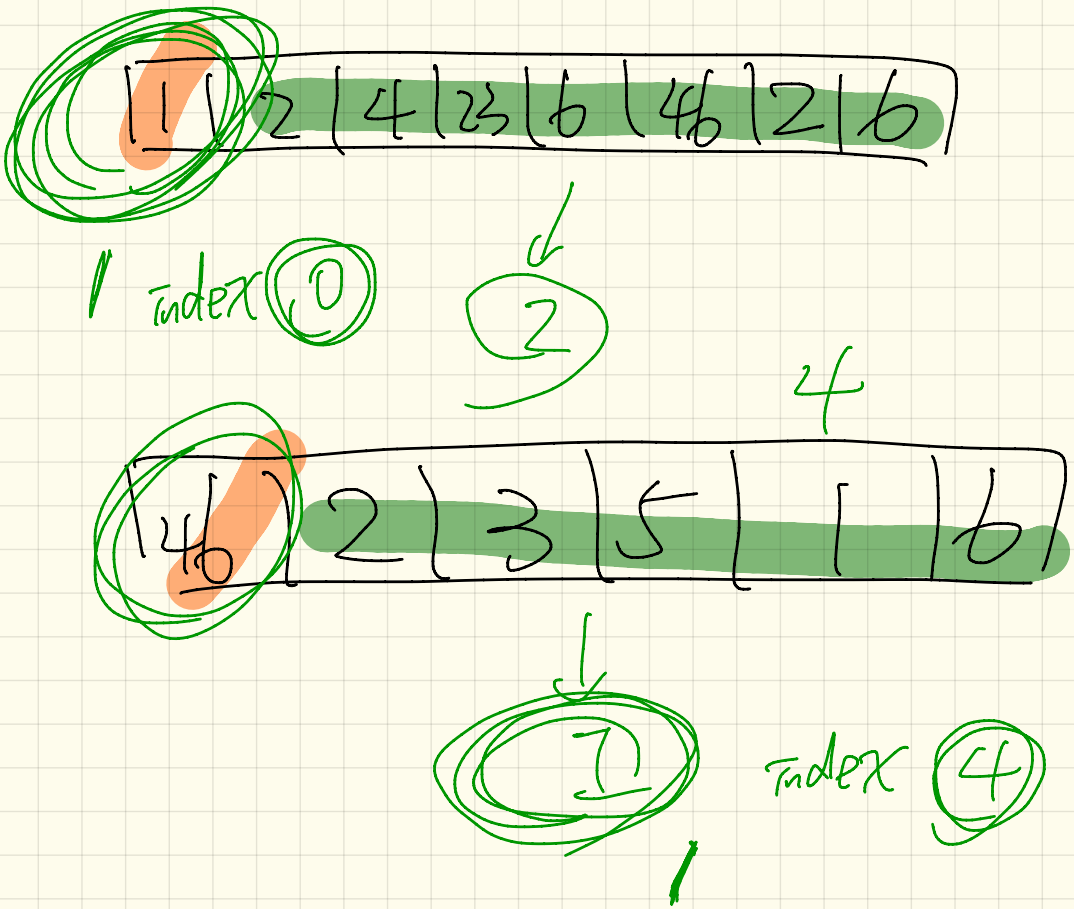
Base Case

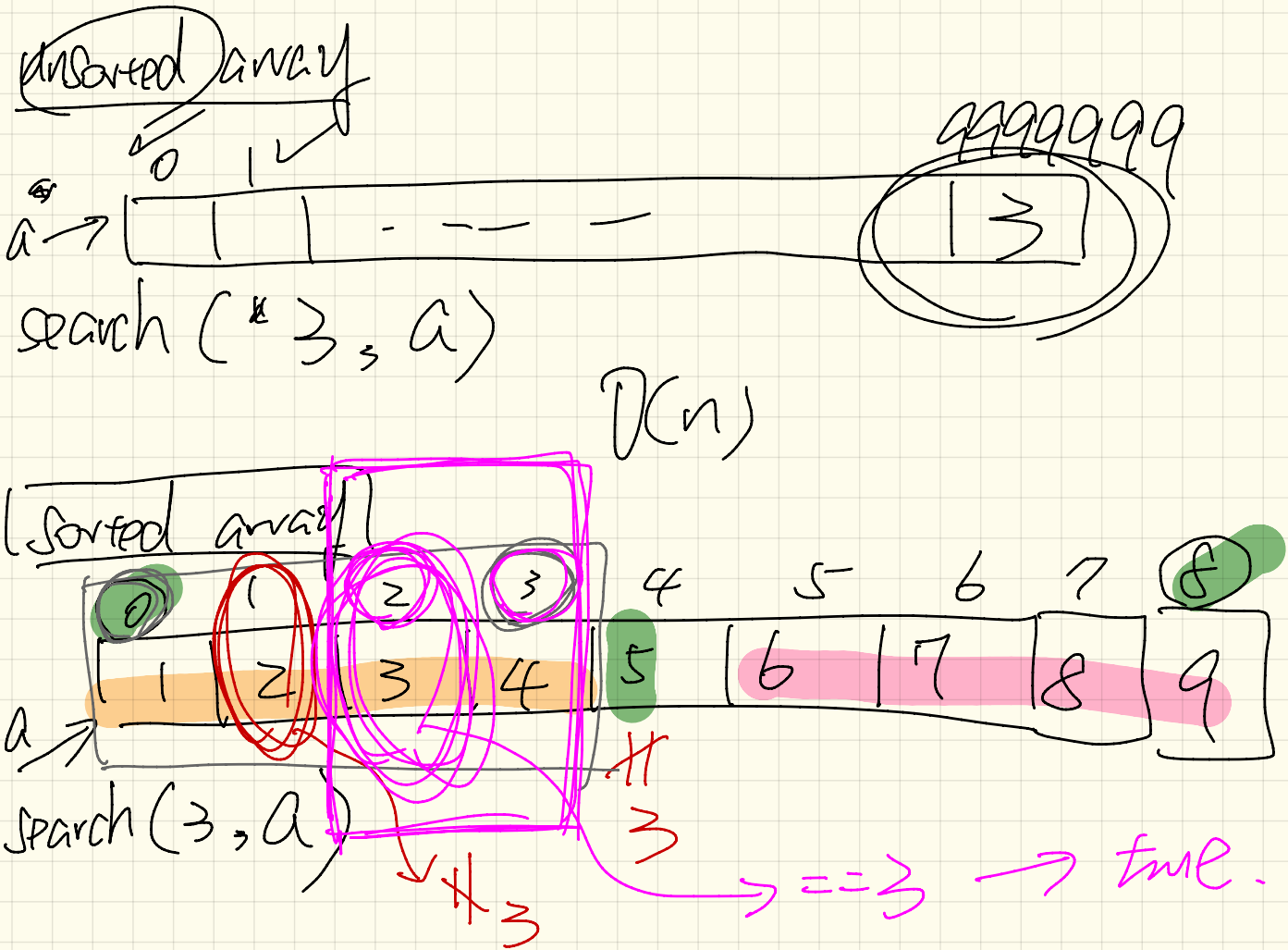
a.length == 1 → return a[0]

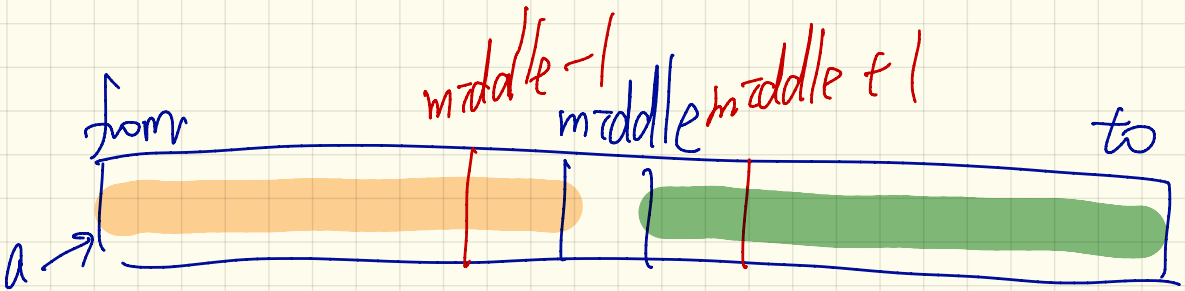


findMin(a, 1, a.length - 1)

↳ min of

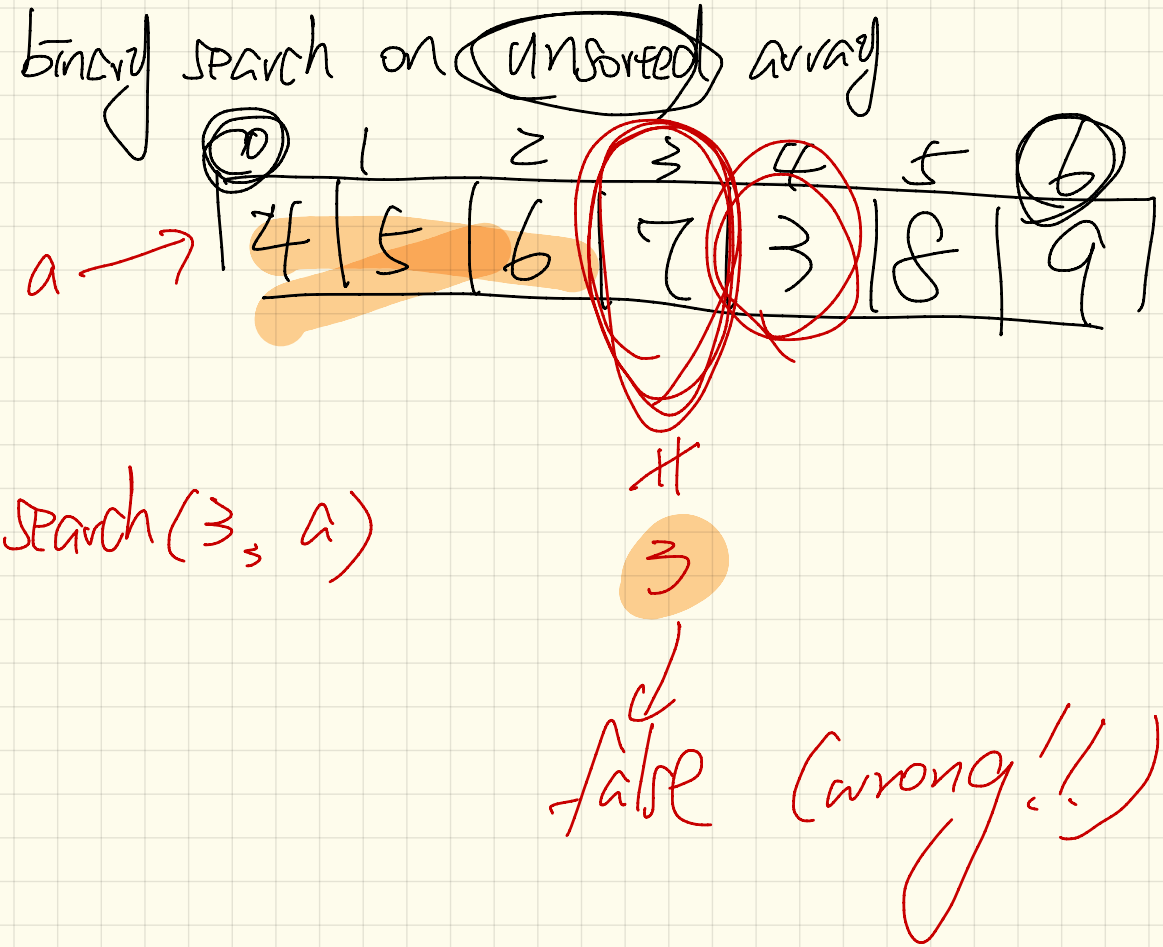

1 2 4 23 6 46 2 6
index 0
2
4
46 2 3 5 1 6
1
index 4


unsorted array
0
1
a
3
search(3, a)
O(n)
sorted array
0
1
2
3
4
5
6
7
8
9
a
search(3, a)
3
3
==3
true.

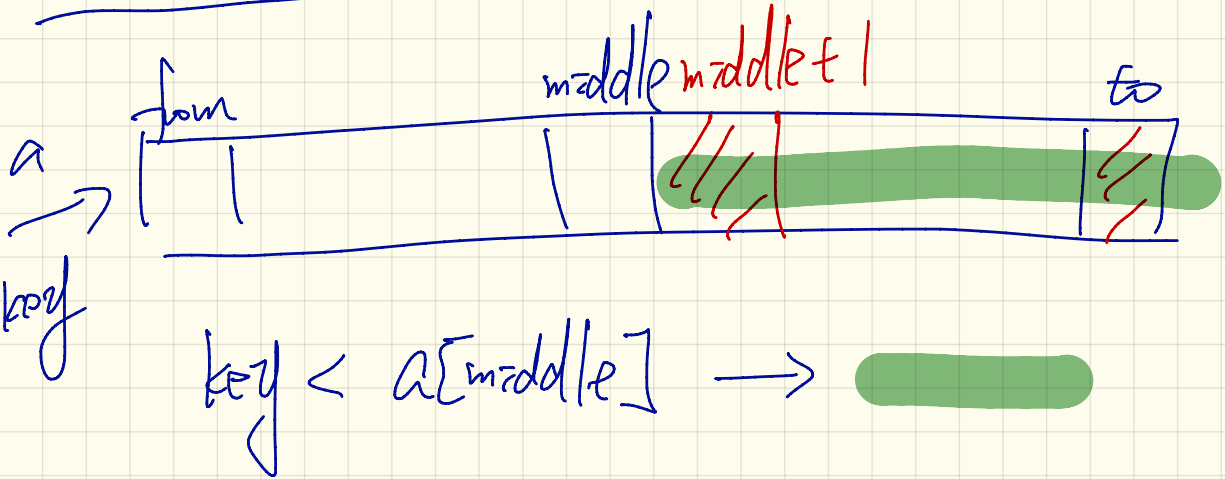


key

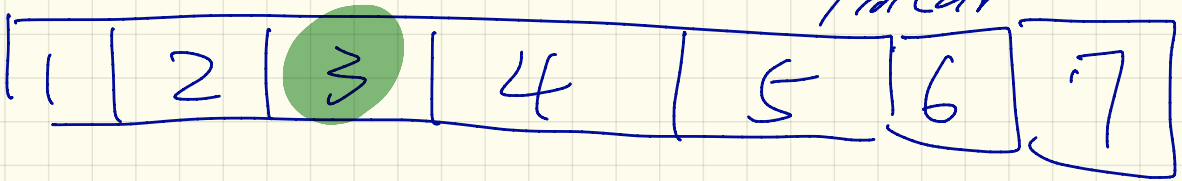
key < a[middle]


binary search on unsorted array
0
1
2
3
4
5
6
4
5
6
7
3
8
9
a
search(3, a)
3
false (wrong!!)

How to do binary search if the array is sorted in descending order?

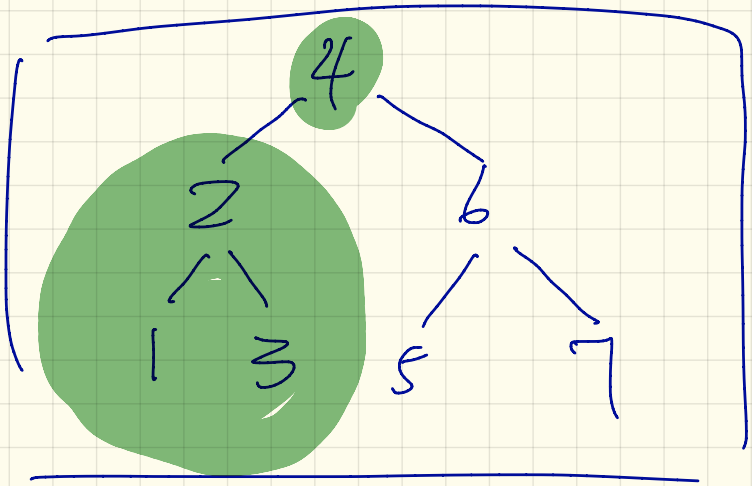


linear



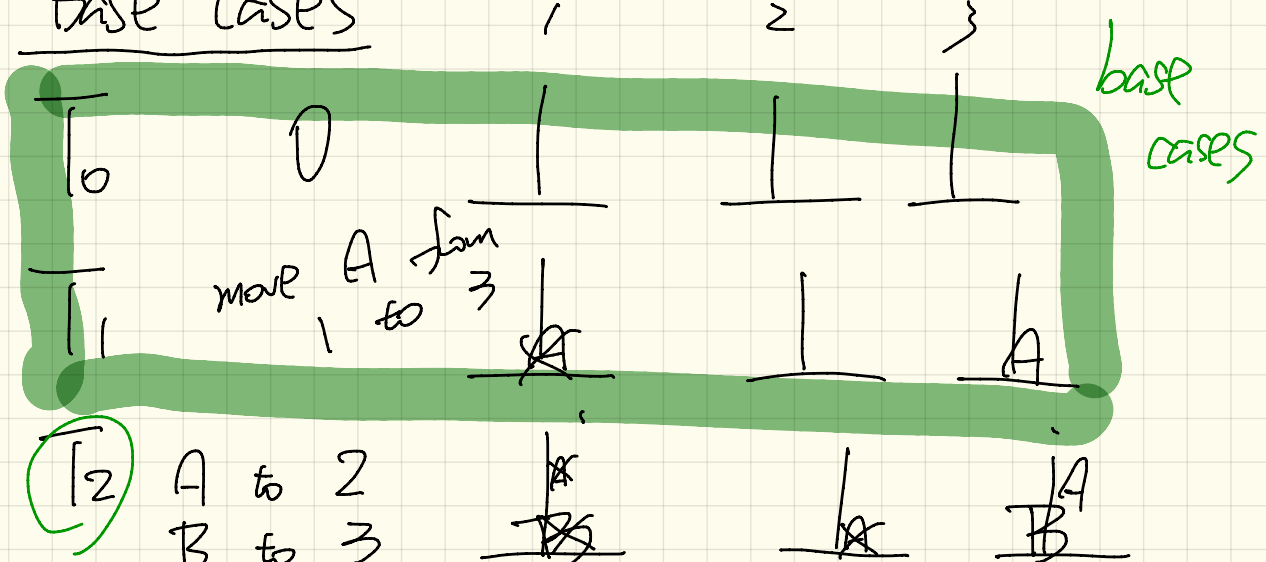
binary-search-tree

non-linear

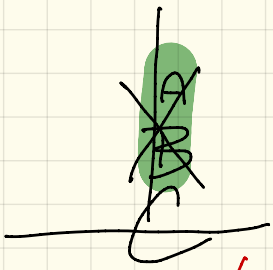


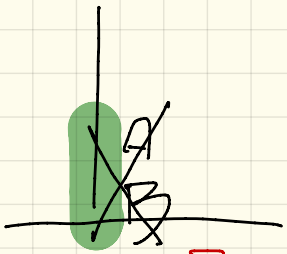
Tower of Hanoi (move from peg 1 to peg 3)

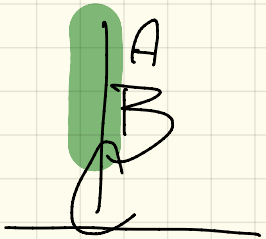
Base Cases



T_3







move A B C from 1 to 3

move $\overset{A}{B}$ from 1 to 2? 3?

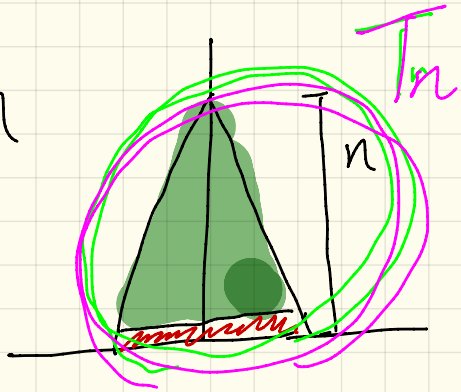
3 Steps:

Move $\overset{A}{B}$ from 1 to 2

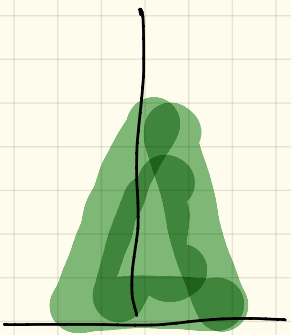
Move C from 1 to 3

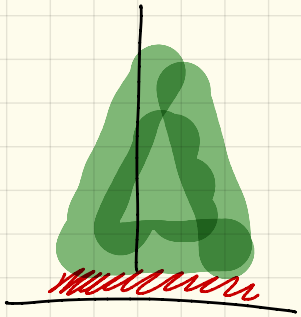
Move $\overset{A}{B}$ from 2 to 3

T_n



T_n





$T_0 = 0$

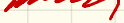
$T_1 = 1$

$T(n) = T(n-1) + 1 + T(n-1)$

$2 \cdot T(n-1) + 1$

3 Steps

✓ - Move  from 1 to 2 T_{n-1}

- Move  from 1 to 3 1

- Move  from 2 to 3 T_{n-1}

- $T(0) = 0$
- $T(n) = 2 \cdot T(n-1) + 1$

$$T(4) = 2 \cdot T(3) + 1$$

$$= 2 \cdot (2 \cdot T(2) + 1) + 1$$

$$= 2 \cdot (2 \cdot (2 \cdot T(1) + 1) + 1) + 1$$