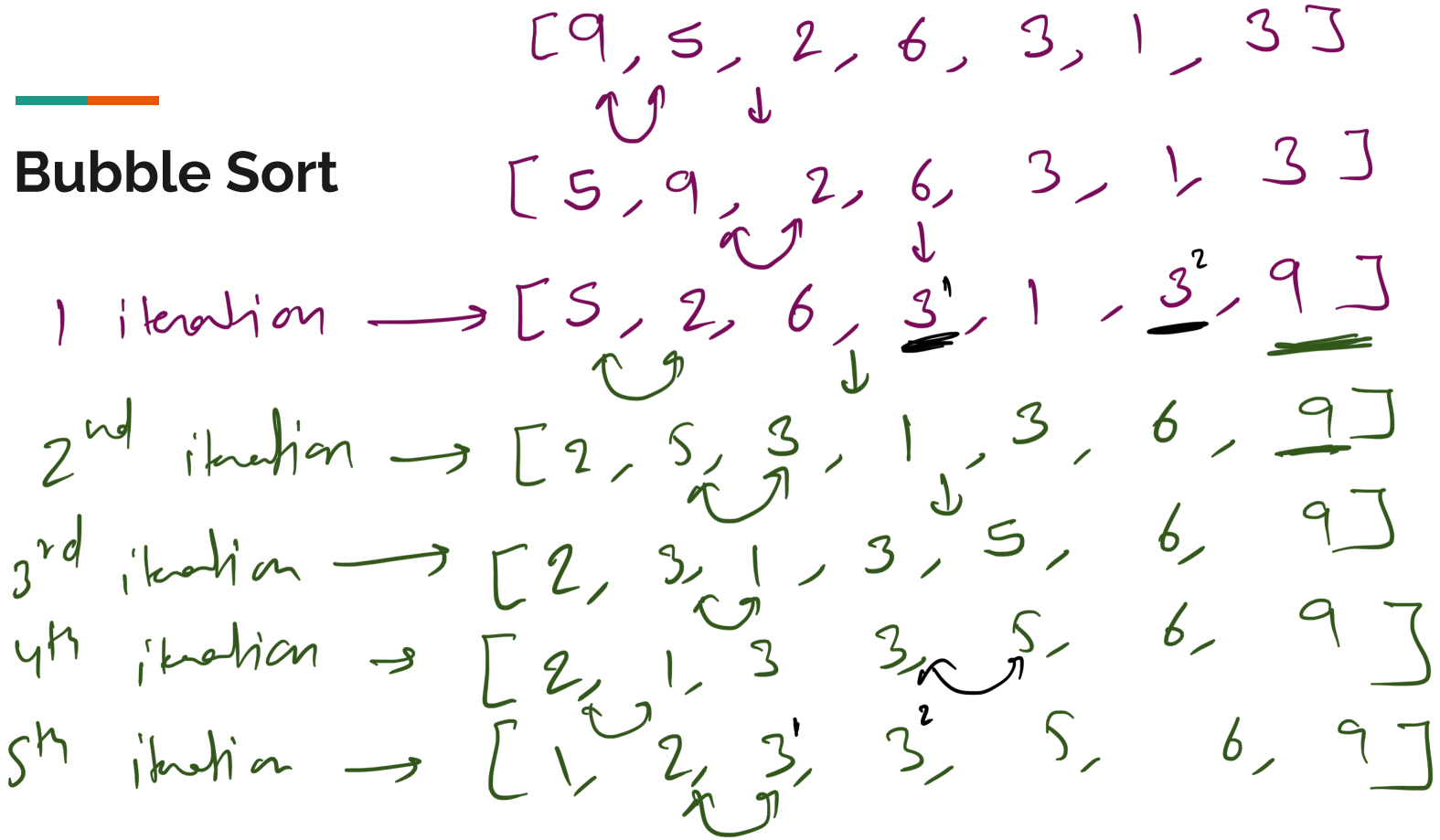
Sorting Algorithm - I

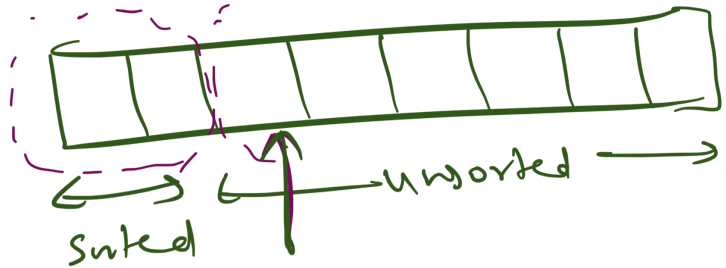
Bubble Sort


[9, 5, 2, 6, 3, 1, 3]
[5, 9, 2, 6, 3, 1, 3]
1 iteration → [5, 2, 6, 3, 1, 3, 9]
2nd iteration → [2, 5, 3, 1, 3, 6, 9]
3rd iteration → [2, 3, 1, 3, 5, 6, 9]
4th iteration → [2, 1, 3, 3, 5, 6, 9]
5th iteration → [1, 2, 3, 3, 5, 6, 9]

Time → $O(N^2)$

Space → $O(1)$

Insertion Sort



a = [9, 5, 2, 6, 3, 1, 3]

temp = 1

1st → [2, 5, 9, 6, 3, 1, 3]

2nd → [2, 5, 6, 9, 3, 1, 3]

3rd → [2, 5, 6, 9, 3, 1, 3]

4th → [2, 3, 5, 6, 9, 1, 3]

i = 5
j = 4

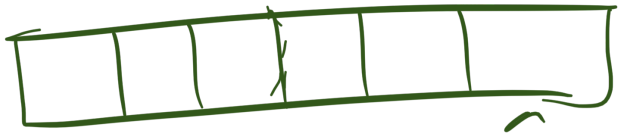
0 1 2 3 4 5 6

5th → [1, 2, 3, 5, 6, 9, 3]

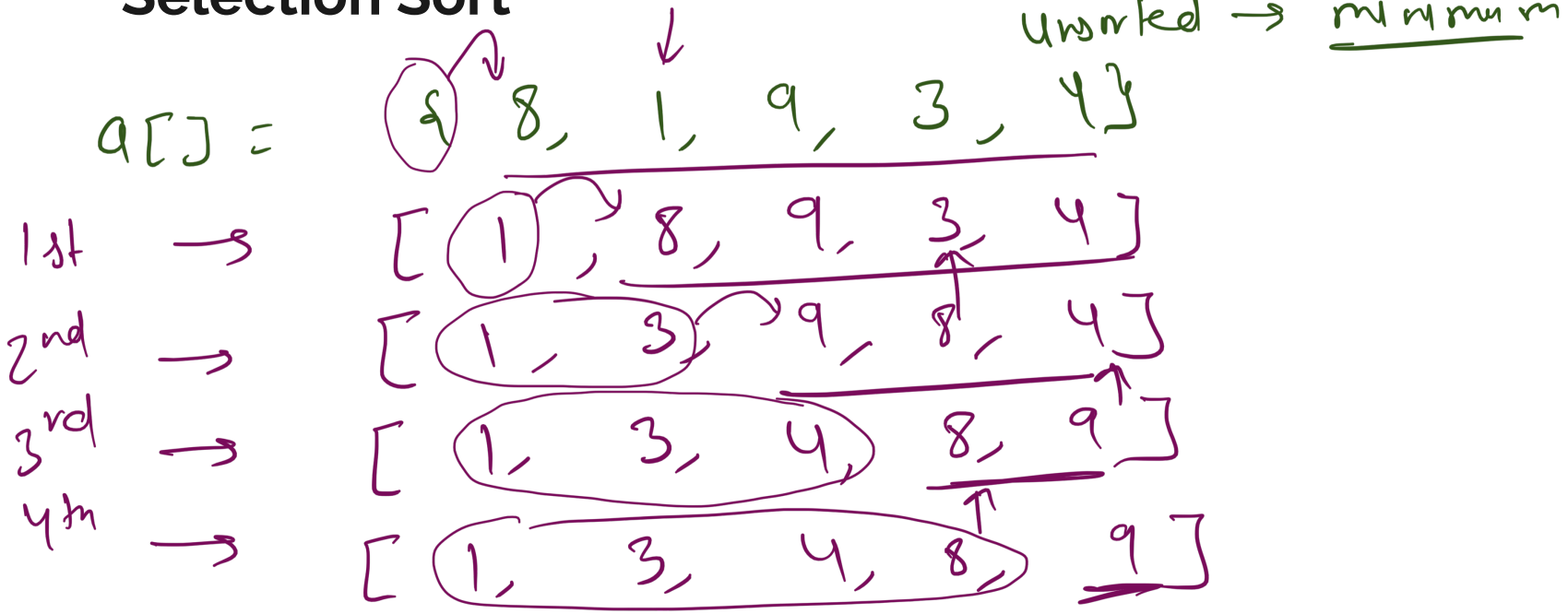
j = -1

T.C. → $O(N^2)$.

Selection Sort



Unsorted → minimum



Stable Sort

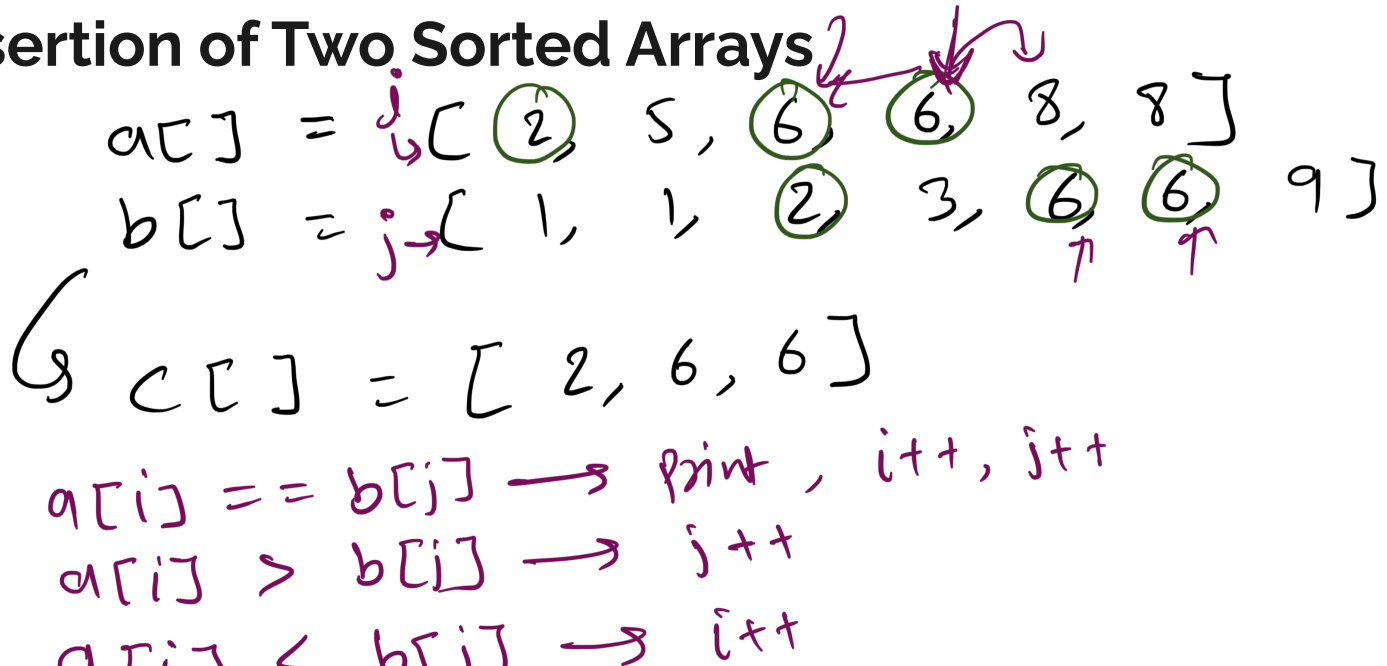
a[] = { 2, 3, 9, 1, 3, 6, 1, 9 }

Sort ↓

{ 1, 1, 3, 3, 6, 9, 9 }

Student
int marks
String Name

Insertion of Two Sorted Arrays



Sort an array with three types of elements.

a[] = [2, 2, 1, 0, 0, 1, 2, 0, 1]

$O(N^2)$

↓

$O(N \log N)$

↓

$O(N)$

iterations → 2

1 iteration → Dutch flag sorting Algorithm

a[] = [2, 2, 1, 0, 0, 1, 2, 0, 1]

mid = 0 | swap | low++ | mid++

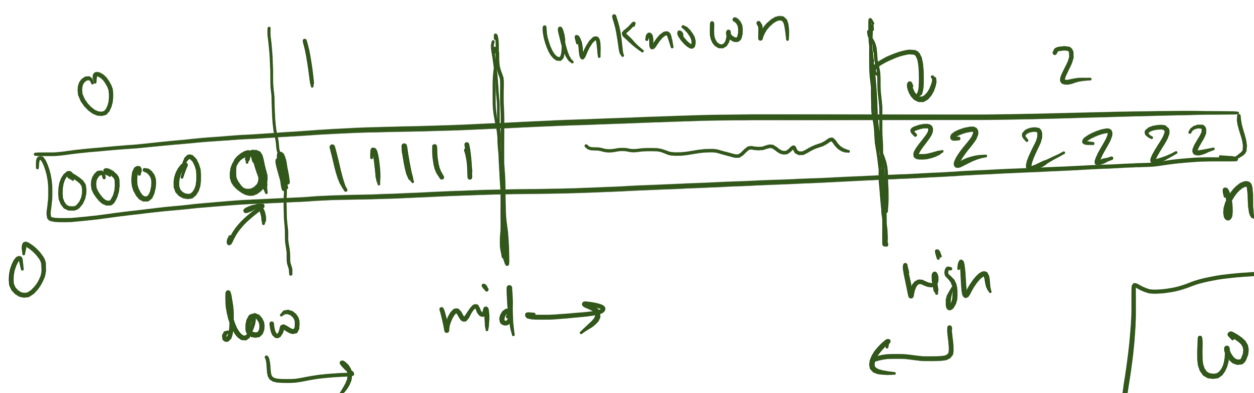
mid = 1 | mid++

mid = 2 | swap(mid, high) | high--

low → 1 inclusive

mid → unknown

high → 2 exclusive



h = n-1

lo = 0

mid = 0

while (mid < high)

Practice Problems

1. Union of Two Sorted Arrays
    a. Union of [1, 2, 2, 3, 5] and [2, 3, 3, 4] is [1, 2, 3, 4, 5]
2. Find the minimum difference between two elements in an array.
    a. Minimum difference is this array: [6, 18, 1, 9, 14] is 3 (because 9-6 = 3)
3. Sort an array of two types of elements.