

Best Case, Worst Case, Average Case

Input

~~No of Inputs (n)~~

n=0

n=1



③



Binary Search



Linear Search

②





Best Case
↓
First element

Worst Case
↓
last element

L.S

A =

K = 7 ✓



for $i = 0$ to $\text{len}(A)$
if $K == A[i]$
break

K = 10^5



Average Case

All possible Cases

1, 2, 3, 4 ...
↑ Best 4 times

↑ Worst times

$$\frac{n(n+1)}{2}$$

$$n$$

$$O\left(\frac{n+1}{2}\right)$$

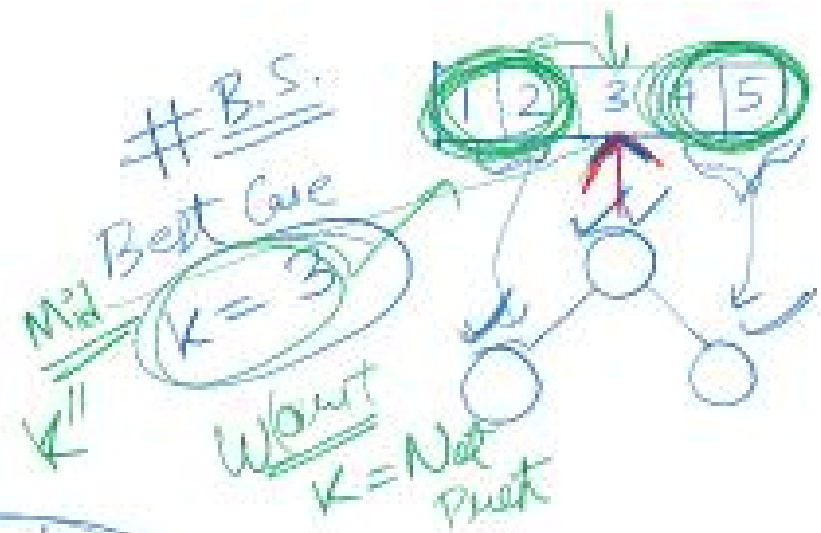
$$O(n)$$

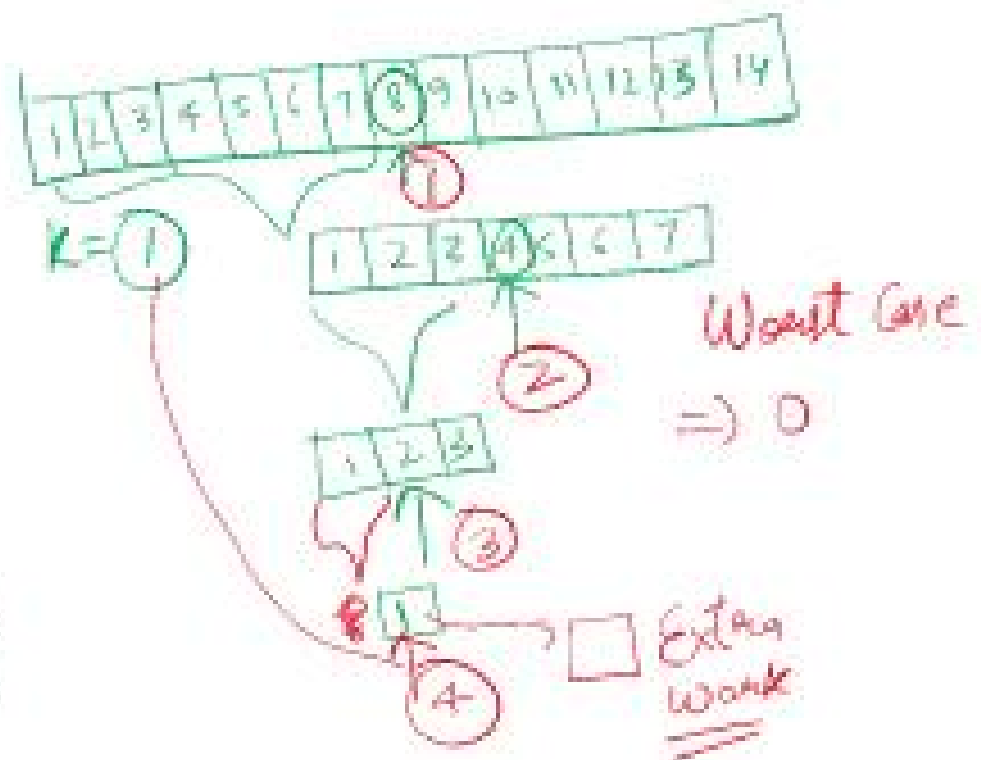
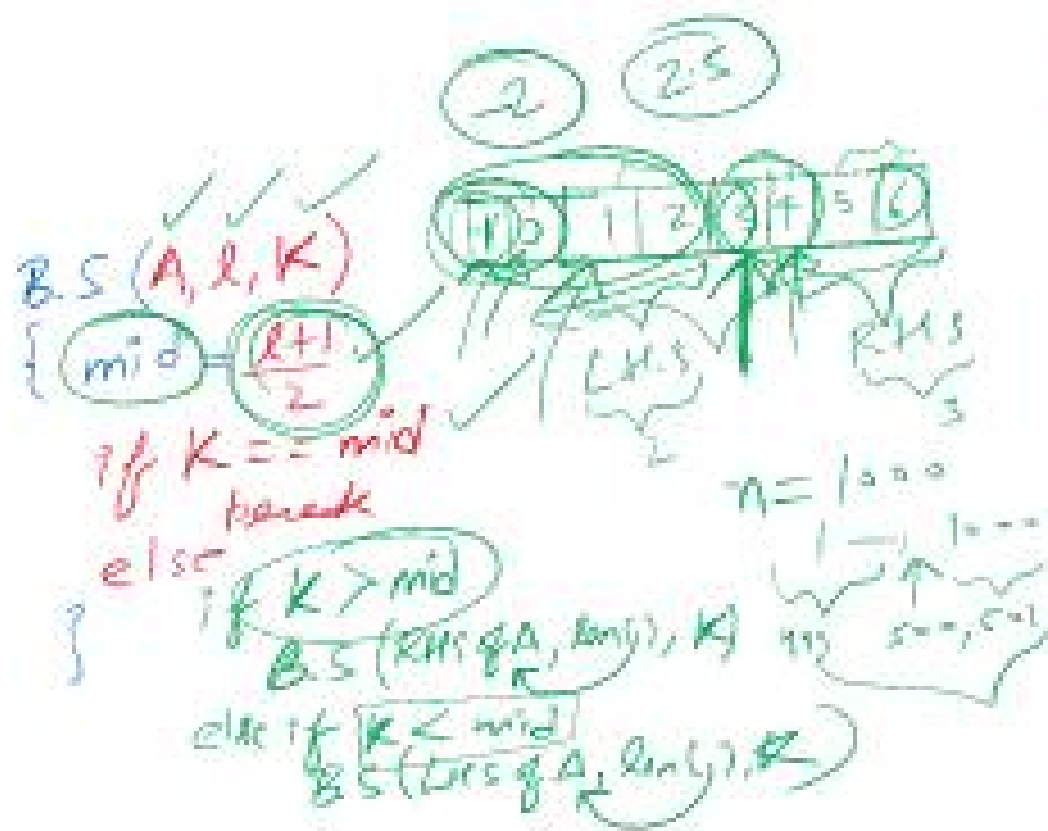
Worst
Case



$n-1$

$n, n-1$





$\{$
 $\text{print}(\text{happy}) \leftarrow O(1)$
 $x = 5 + 3 \leftarrow \text{Assign } O(1)$
 $\text{fun}(x) \leftarrow \text{Calling } O(1)$
 $\{ \text{for}$
 $\}$
 $\}$

$\text{for } i = 0 \text{ to } n$
 $\{$
 $x = 5 + 3$
 $O(1) \times n$
 $1 + 1 + 1 + \dots$
 $\}$
 $n \text{ times.}$
 $O(n)$

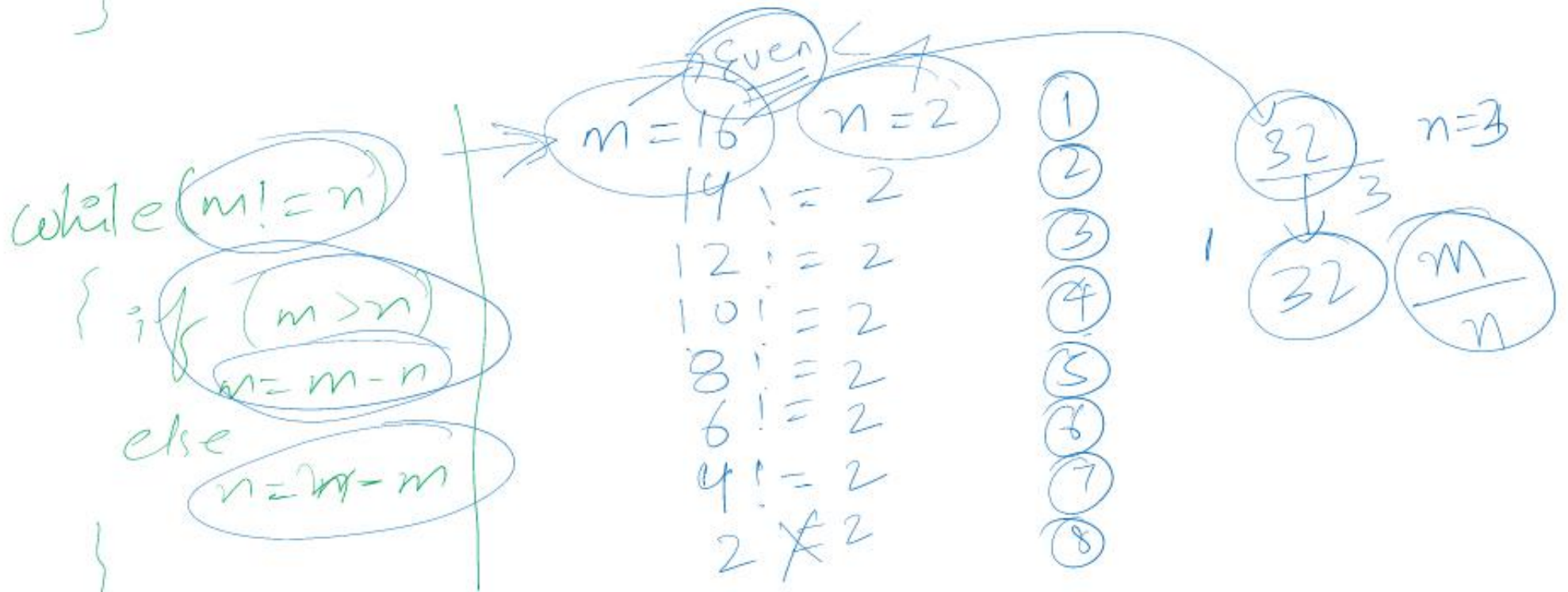
$\text{for } i = 0 \text{ to } n$
 $\{$
 $\text{for } j = 0 \text{ to } n$
 $\{$
 $x = 5 \text{ — } O(1)$
 $\}$
 $\}$
 $\}$

$i = 0 \quad 1 \quad 2 \quad 3 \quad \dots \quad n$
 $\Rightarrow n + n + n + n + \dots + n$
 $\Rightarrow n \times n$
 $O(n^2)$

Q1. $\text{while } (m \neq n)$
 $\{$
 $\text{if } (m > n)$
 $\quad m = m - n;$
 else
 $\quad n = n - m;$
 $\}$

Q1. while (m != n)
 {
 if (m > n)
 m = m - n;
 else
 n = n - m;
 }

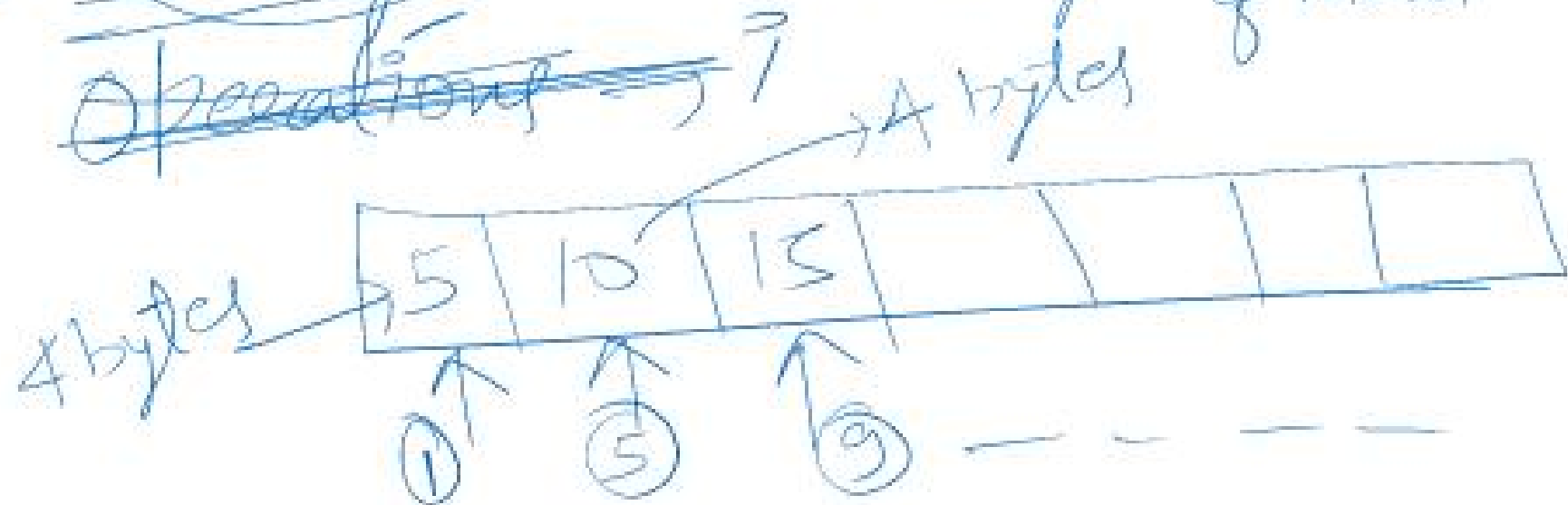
for i = 0 to n i = 0 1 2 3 ... n
 = m + m + m + m + ... m
 = n * m
 {
 for j = 0 to m
 {
 x = 3 — O(1)
 }
 } O(m)



D.S. — ADT

Python
~~list~~ []

① Arrays (homogeneous)
(contiguous block of Mem)
~~Operations~~ → 7



D.S

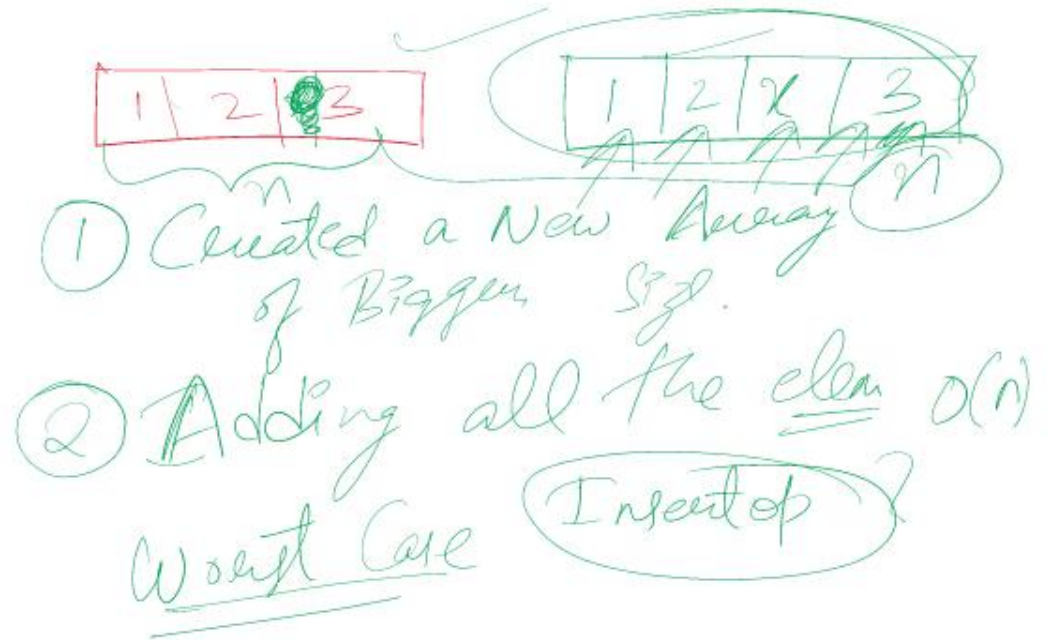
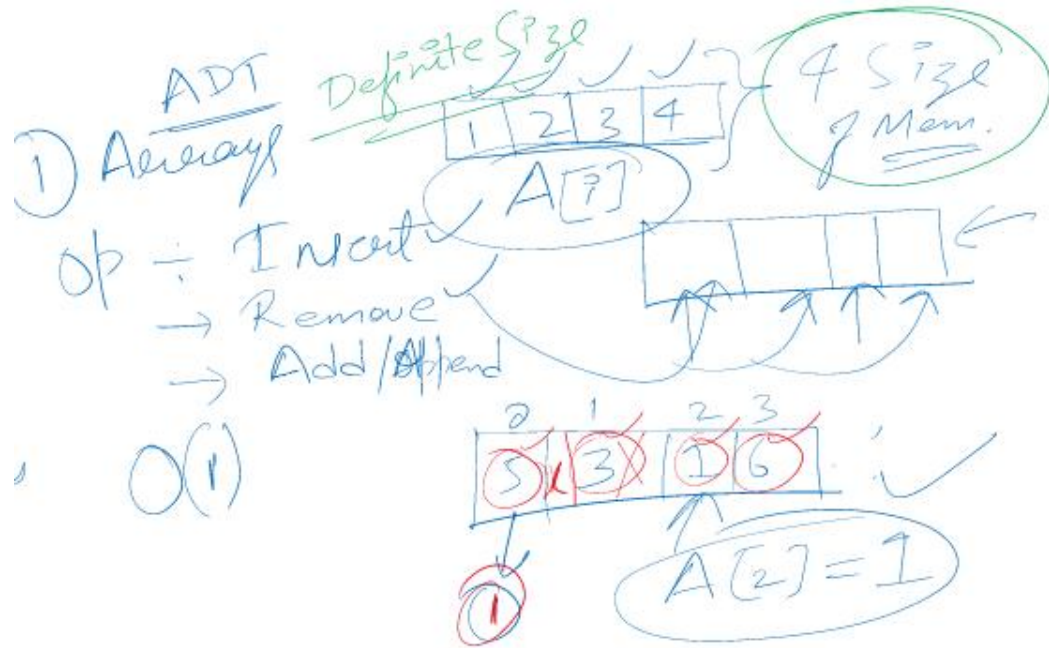
① Arrays
Operations

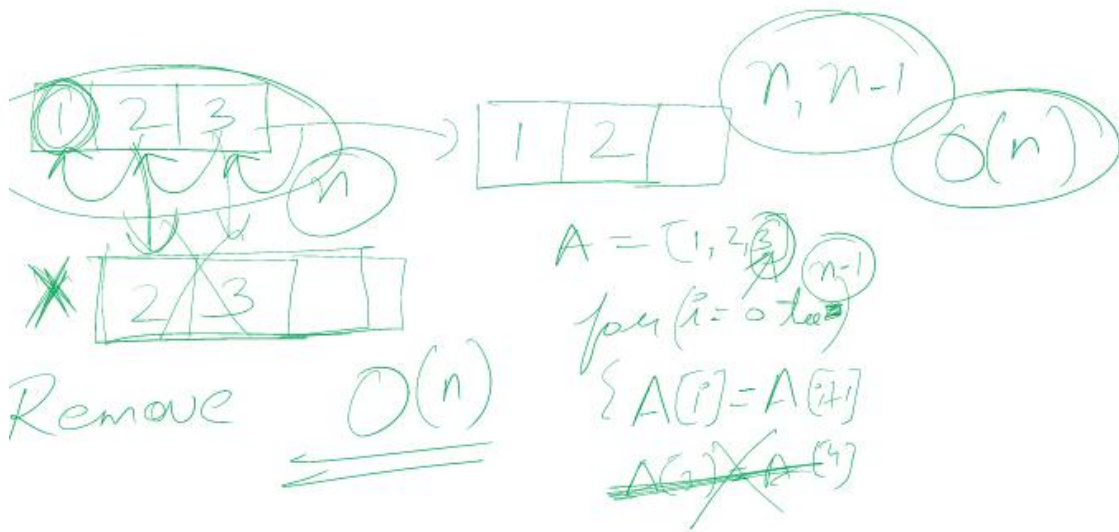
→ Class Array

{ method/functions

Constructor (/ data members

{ arr. Value = []
}
}





class Array {
 constructor(n) {
 // size of array
 // init
 obj = Array(-1)
 }
 // PP
 def Insert ~~Insert()~~ ~~Insert()~~
 {
 }
 Remove
 Access[i]

Homogeneous

Homework

None/None

4 Units

1	None	None	None
---	------	------	------

1950

4 C Prog

$\times 4 = 16$ Units

Balance Array Vs list Array