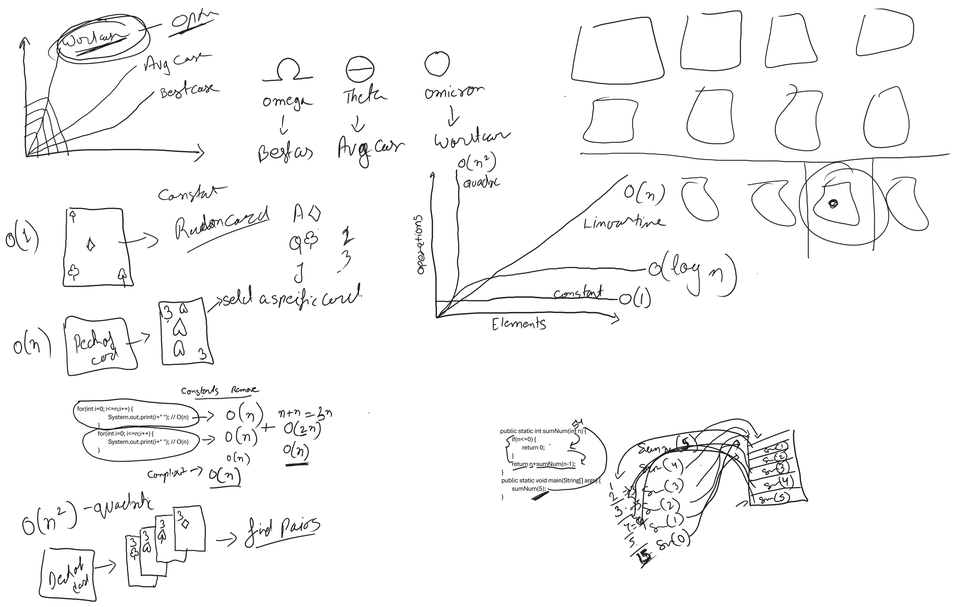
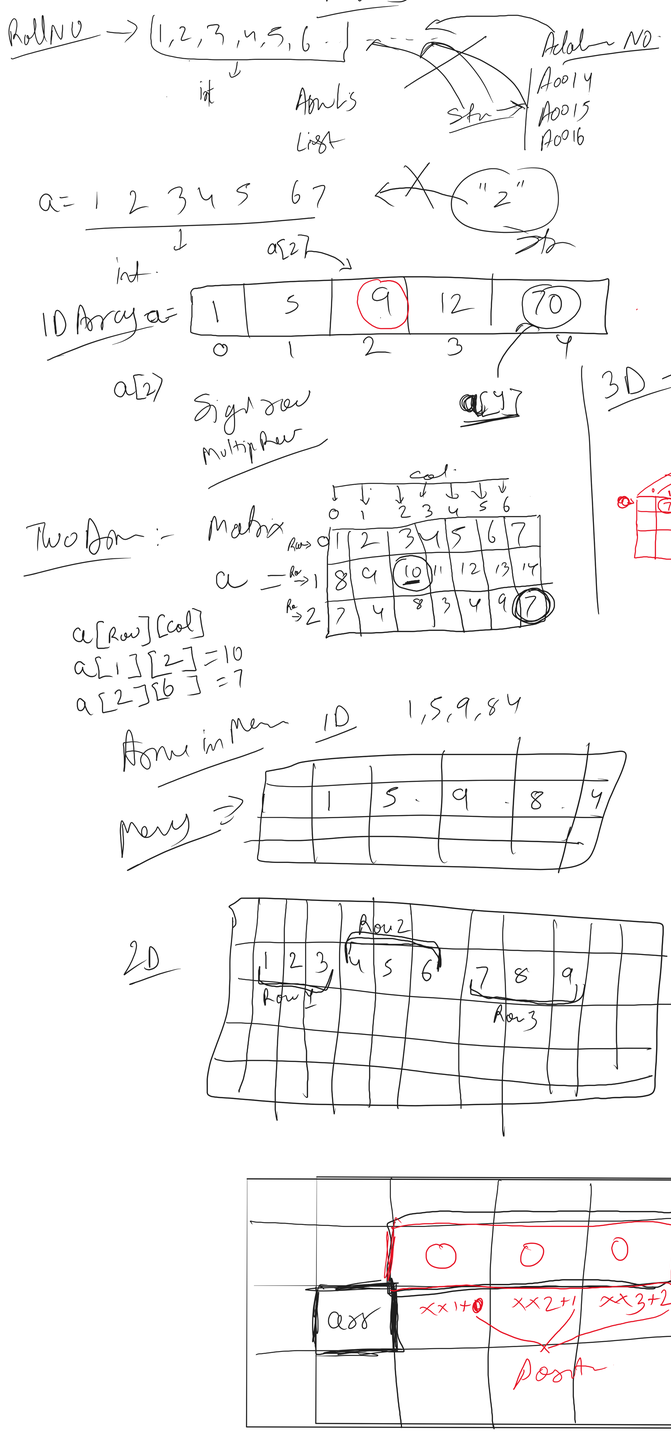
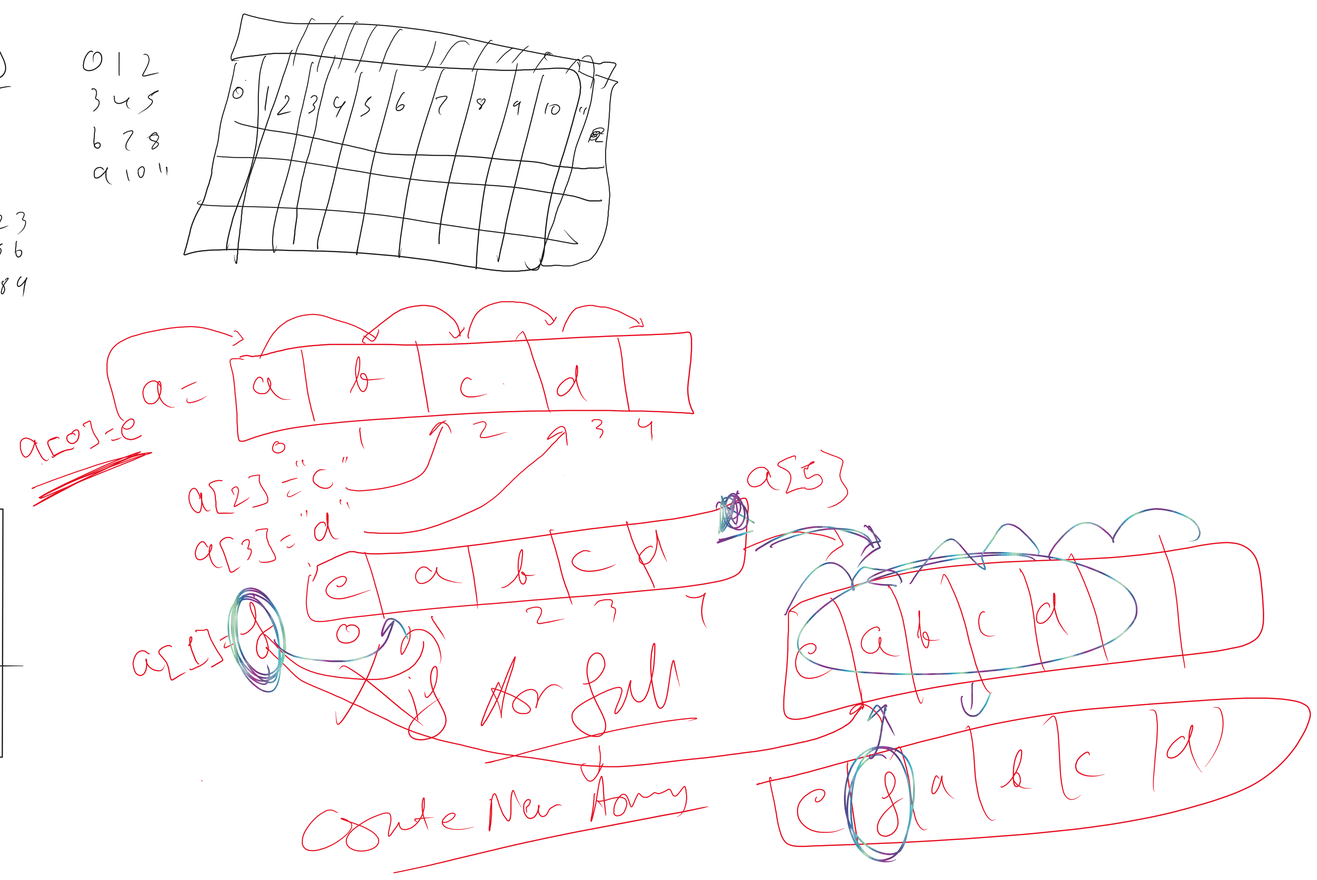

Worst case
Opt
Avg case
Best case
omega
Theta
omicron
Best case
Avg case
Worst case
O(n²)
quadratic
operations
Elements
O(n)
Linear time
O(log n)
constant
O(1)
Constant
O(1)
Random card
select a specific card
O(n)
Redroy card
Constants Remove
O(n)
O(n)
O(n)
Complint → O(n)
O(n²) - quadratic
Deck of card
find Pairs
public static int sumNum(int n){
if(n<=0) {
return 0;
}
return n+sumNum(n-1);
}
public static void main(String[] args){
sumNum(5);
}
sum(5)
sum(4)
sum(3)
sum(2)
sum(1)
sum(0)
15

Array


RollNO → 1,2,3,4,5,6
int
Array
List
Str
Actual NO.
A0014
A0015
A0016
a = 1 2 3 4 5 6 7
"2"
int
a[2]
1D Array a =
1 5 9 12 70
0 1 2 3 4
a[2]
a[4]
Single row multiplier
Two Dim :- Matrix
a =
col
Row
1 2 3 4 5 6 7
8 9 10 11 12 13 14
7 4 8 3 4 9 7
a[Row][col]
a[1][2] = 10
a[2][6] = 7
3D →
a=[Row][col][Depth]
a=[0][0][1]
a=[2][0][2]
Array in Mem
1D
1,5,9,8,4
Mary →
1 5 9 8 4
3D
0 1 2
3 4 5
6 7 8
9 10 11
2D
Row2
1 2 3 4 5 6 7 8 9
Row1
Row3
1 2 3
4 5 6
7 8 9
arr
xx1+0
xx2+1
xx3+2
Depth


a = a b c d
0 1 2 3 4
a[0]=e
a[2]="c"
a[3]="d"
a[5]
e a b c d
0 1 2 3 4
a[1]=f
if Arr full
Create New Array
e a b c d
e f a b c d