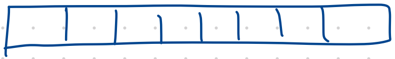
Doubts

→ Additional Class   Doubts + Extra Questions

ARRAYS



↓

Data (Primitives / objects)

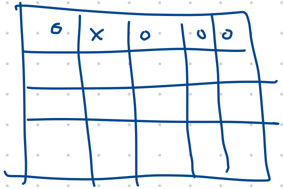
- Insert
- Search
- Delete

Search → Linear Search (Unsorted) N

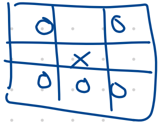
Search → Binary Search (Sorted) Log N

Problems:

chess

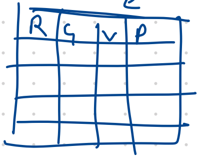


Tic-Tac-Toe

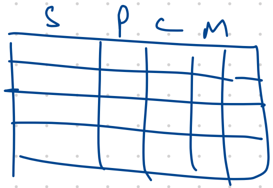


Images

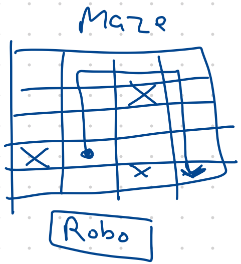
↙ matrix of Pixels



Excel sheet



Graphs



How do create 2-D Array (Matrix) ?

int [] oneDArr =

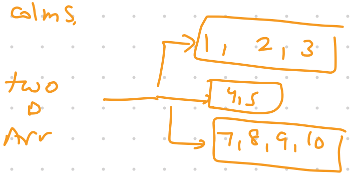
= { 1,2,3,4,5 - - - } ;

int[]

int [ ][ ] twoD Arr = { {1,2,3}, {4,5}, {7,8,9,10} };

↳ Array where every element is an Object (array

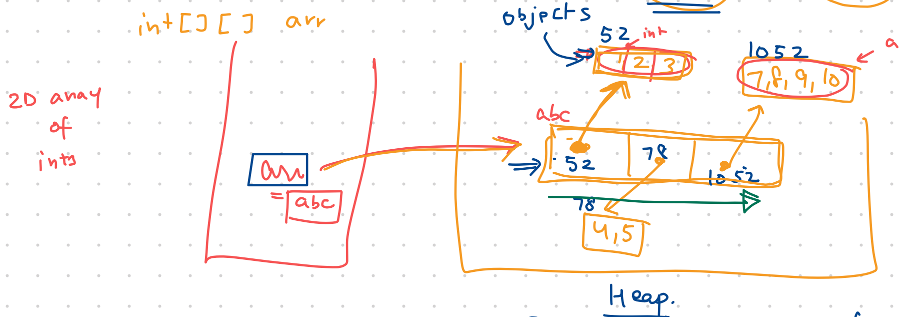
⇒ variable No of colms



Array of Objects

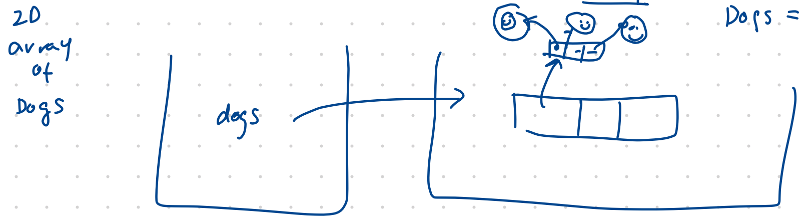
int[][] arr

arr = { $\overset{57}{\{1,2,3\}}$, $\overset{78}{\{4,5,6\}}$, $\overset{1052}{\{7,8,9\}}$ };



arr[i] length ✓

arr[i][j] ✓

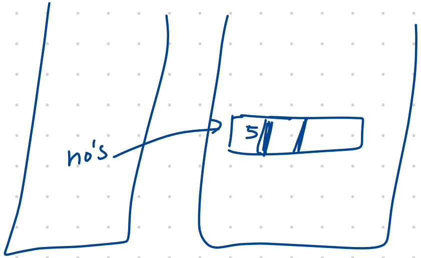


Dogs = { {d1,d2,d3},

{d4,d5} }

Linear

1D Array
of ints

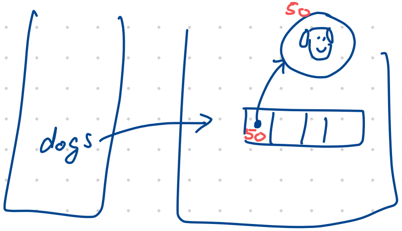


nos[0] = 5

d1 = new Dog();

1 D of
dogs.

dogs = {d1,d2,d3}

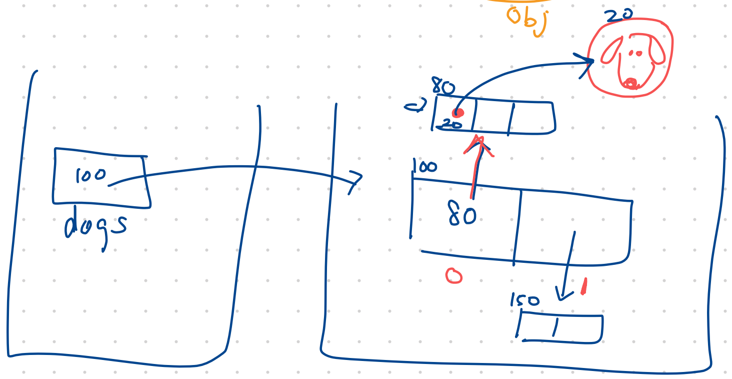


dogs[0] = d1 = new Dog().

obj

Dogs [ ] [ ] dogs = { {d1, d2, d3},
{d4, d5} },

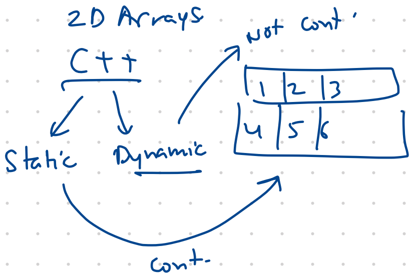
obj



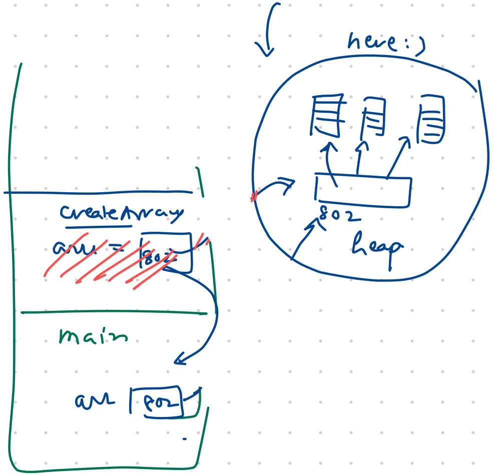
1D arrays → conti

2D arrays → X

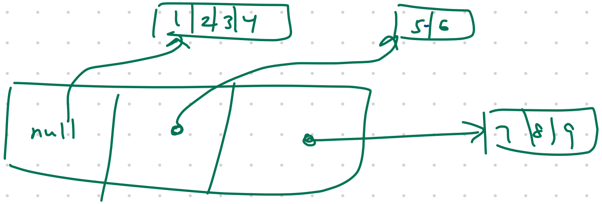
dogs[0][0] = new Dog();



```
createArray() {

    arr()() —> new arr(row)(cols)


  }


main() {

   int[][] arr = createArray()


  }
```

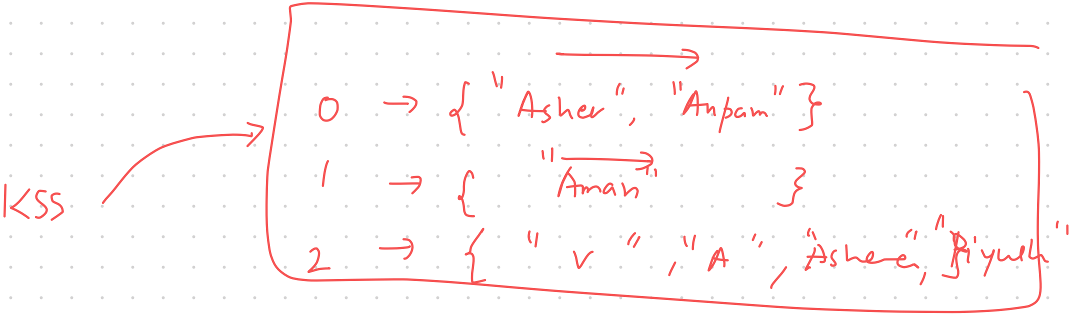


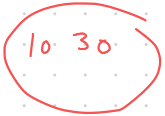
int[][] arr = new int [rows][ ];

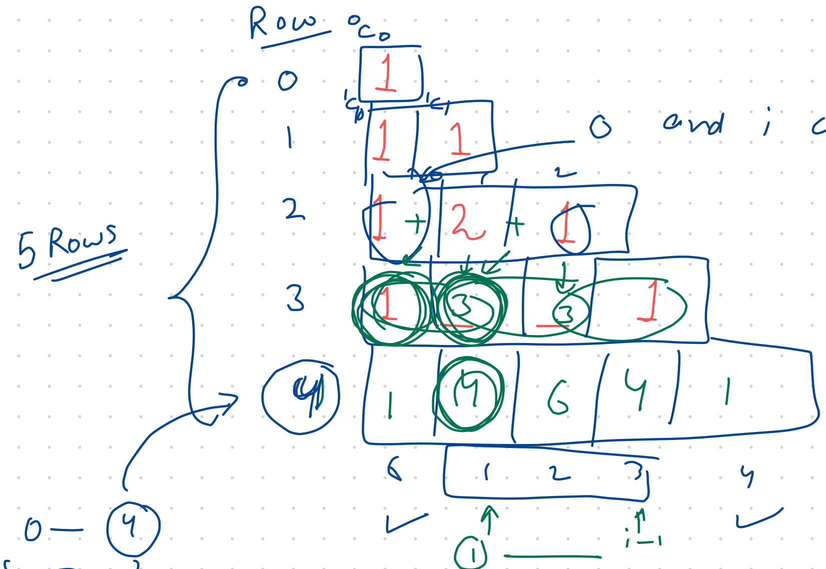


arr [0] = $\underbrace{\text{new int [4]}}$;

1, 2, 3, 4


KSS
0 → { "Asher", "Anpam" }
1 → { "Aman" }
2 → { "v", "A", "Asher", "Piyush" }


10 30



0 and i col => 1

i = Row → i+1 cols

arr[i][j] = arr[i-1][j] + arr[i-1)(j-1)

0th Row   1

1   1

1   2   1

1   3   3   1

5 Rows

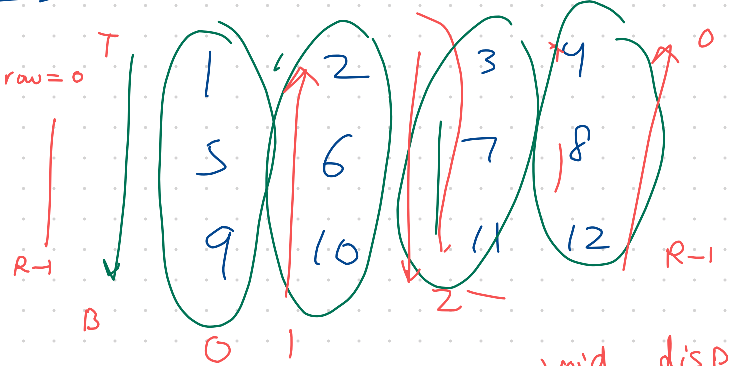
$\boxed{4}$ 1   4   6   4   1

$$\begin{tikzcd} {}^4C_0 \arrow[r, bend right] & {}^4C_1 \arrow[r, bend right] & {}^4C_2 \arrow[r, bend right] & {}^4C_3 \arrow[r, bend right] & {}^4C_4 \end{tikzcd}$$

$$\boxed{{}^NC_R = \left(\frac{N-R+1}{R}\right) {}^NC_{R-1}}$$ ✓

Time To TRY   Wave Traversal

cols → fixed



output ⇒ 1, 5, 9, 10, 6, 2, 3, 7, 11, 12, 8, 4

```
void display (int[][] arr) {

}
```

?

"ARRAY List"

- ↳ More powerful than array
- ↳ Dynamically grow & shrink in size

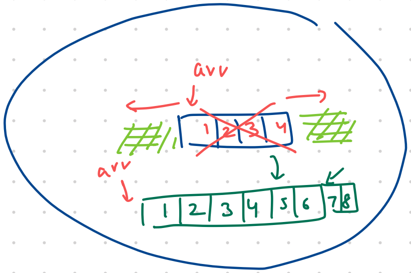
Java → Collections framework

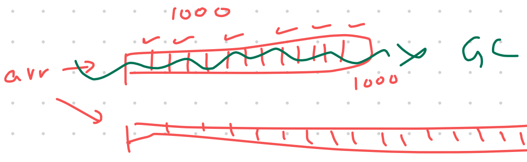
⇓

Data Structures

⇓

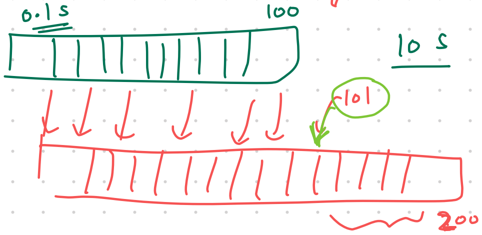
[Ready To use]





Slow when Re-allocation

101th element



10 s

$100 \rightarrow \boxed{10s}$

+

$10s + 10s \rightarrow \boxed{20s}$

$= 30s.$

You should set init that s.t you require element

Avg time per = 0.1 s.

$= 0.15s \left(\frac{30}{200}\right)$

$\boxed{1000}$