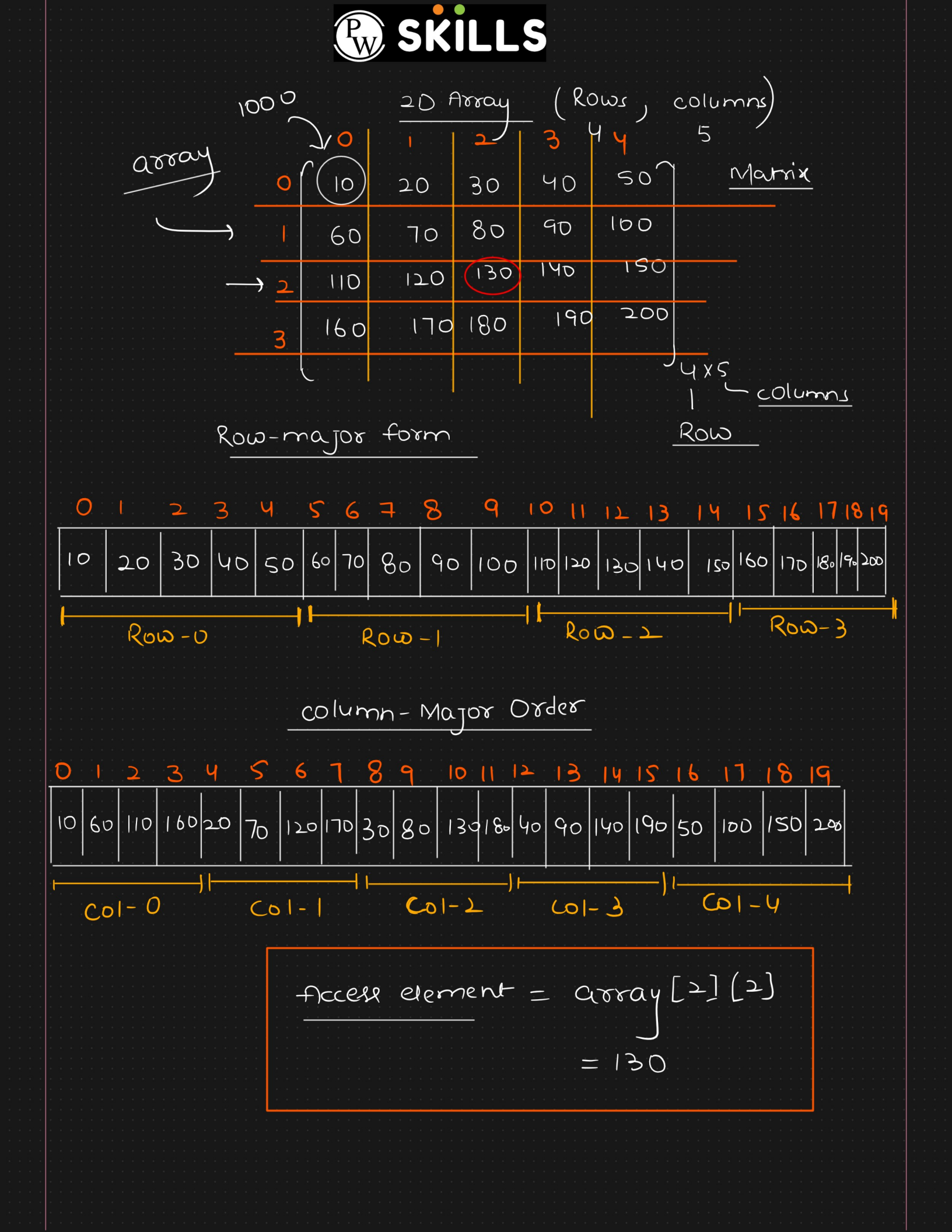
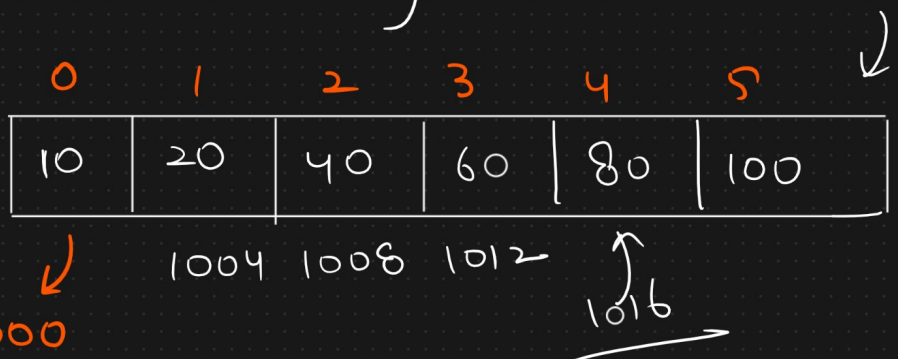

1000
2D Array (Rows, columns)
4 5
array
Matrix
0 1 2 3 4
0 10 20 30 40 50
1 60 70 80 90 100
2 110 120 130 140 150
3 160 170 180 190 200
4 X 5
columns
Row-major form
1 Row
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
Row-0
Row-1
Row-2
Row-3
column-Major Order
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
10 60 110 160 20 70 120 170 30 80 130 180 40 90 140 190 50 100 150 200
Col-0
Col-1
Col-2
Col-3
Col-4
Access element = array[2][2]
= 130

Addressing of an element

1D Array



int = 4 Bytes

i = 4

Address of 4th index element = 1000 + (4 − 0) * 4

= 1000 + 16

= 1016

BA + (i − LB) * Size

2D Array

mR = 4, mC = 5

i = 2, J = 2

Row major form

1000 + ((2 − 0) * 5 + (2 − 0)) * 4

1000 + (12) * 4

1060

BA + ((i − LBr) * mC + (J − LBc)) * size

Column-major form

$i \rightarrow J$

$LBr \rightarrow LBc$

$nR \rightarrow nc$

$$BA + \left(\left(J - LBc \right) * nR + \left(i - LBr \right) \right) * \text{size}$$

Implementation 2D Array

$$\begin{cases} m = 4 \\ n = 5 \end{cases}$$

↳ Total elements $= 4 * 5$

$\Rightarrow 20$

```
for(i=0; i<m; i++){
    for(j=0; j<n; j++){
        print(arr(i)(j));
    }
}
```

$O(m * n)$