2D Array - 3

2D Vectors

Lecture- 16

Raghav Garg

2D Vectors

- `vector<vector<int> > v;`
- `vector<vector<int> > v(m);` → initial size
- `vector<vector<int> > v(m, vector<int> (n));`
- `vector<vector<int> > v(m, vector<int> (n, k));`

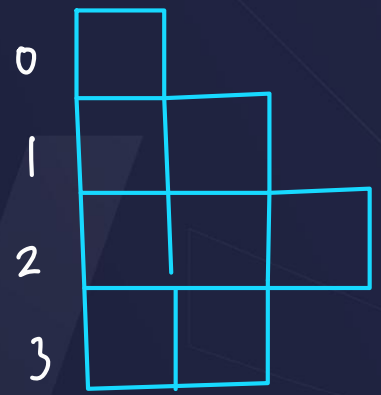
2D

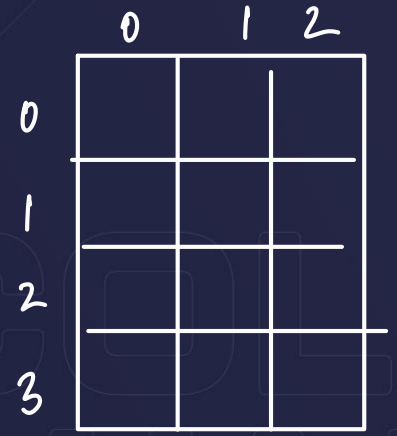
Advantages of vectors over arrays

1) Increase your rows

2) Variable Columns

2D vectors are vector of vectors



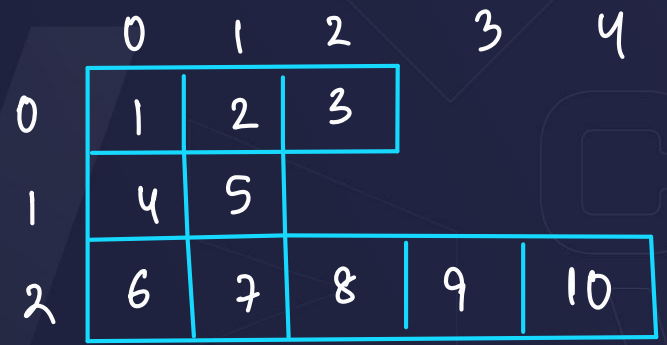


Advantages of vectors over arrays

v1 = {1, 2, 3} v2 = {4, 5} v3 = {6, 7, 8, 9, 10}

v = { {1,2,3} , {4,5} , {6,7,8,9,10} }

0 1 2




```
cout << v[1][1];
```

Advantages of vectors over arrays

Passing of 2D Arrays into functions → problem

we have to mention rows & columns both

2D Vector as 2D Arrays

Basic STL functions in vectors

- push_back()
- pop_back()
- begin()
- end()
- rbegin()
- rend()

v = {{1,2,3}, {4,5}}

a = {6,7,8,9,10}

v.push_back(a);

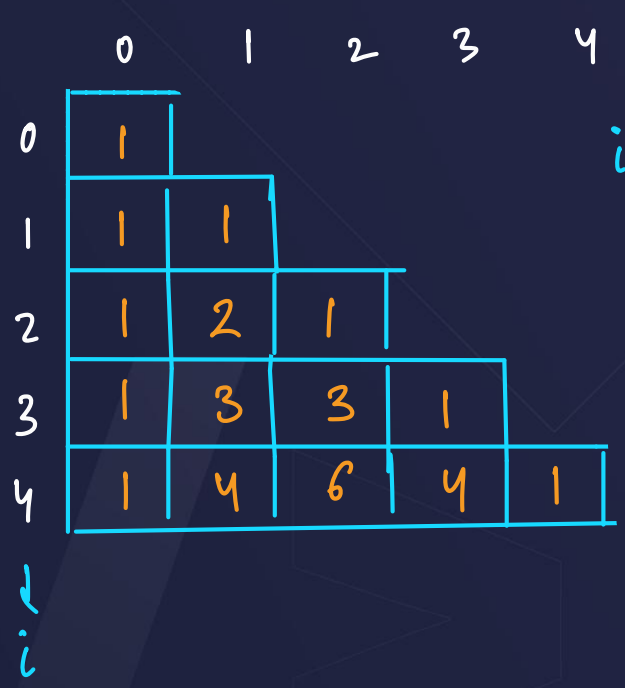
v → {{1,2,3}, {4,5}, {6,7,8,9,10}}

v.pop_back();

v → {{1,2,3}, {4,5}}

Ques : Given an integer 'numRows', generate Pascal's triangle. [Leetcode 118]

5



$$^{i}C_{j} = \frac{i!}{j! \times (i-j)!}$$

```
if(j==0 || j==i) v[i][j]=1;
else
v[i][j] = v[i-1][j] + v[i-1][j-1];
```

Ques : Score after flipping matrix

[Leetcode 861]

	0	1	2	3
0	0	0	1	1
1	1	0	1	0
2	1	1	0	0

sum = 0

1 1 0 0

1 0 1 0

1 1 0 0

1st step: 0th column → all ones → rows flip

↓

initially 0

2nd step: Flip the columns where noz > noo

Binary to Decimal

8	4	2	1
2^3	2^2	2^1	2^0
3	2	1	0
0	0	1	1

X

$= 8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1$

$= 0 + 0 + 2 + 1 = 3$

$0\ 0\ 1\ 1 \rightarrow 3$

16	8	4	2	1
2^4	2^3	2^2	2^1	2^0
4	3	2	1	0
1	0	1	0	1

$\Rightarrow 16 \times 1 + 8 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 1$

$= 16 + 0 + 4 + 0 + 1$

$= 21$


```
int sum = 0;
int x = 1;
for(int i = 6 ; i >= 0 ; i--){
    sum += arr[i] * x;
    x *= 2;
}
```

x = 1 2 4 8 16 32 64 128

Sum = 0 1 1 5 5 5 37

101

No. of ones ↑ → number ↑ → not always true

1	1	0	0	0	0
5	4	3	2	1	0
32	16	8	4	2	1

$1 \times 32 + 1 \times 16 + 0 \times 8 + 0 \times 4 + 0 \times 2 + 0 \times 1$

$= 32 + 16 = 48$

1	0	1	1	1	1
5	4	3	2	1	0
32	16	8	4	2	1

$1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1$

$= 32 + 0 + 8 + 4 + 2 + 1$

$= 47$

Ques : Write an efficient algorithm that searches for a value target in an m x n integer matrix which has the following properties :

- Integers in each row are sorted in ascending from left to right.
- Integers in each column are sorted in ascending from top to bottom.

[Leetcode 240]

	0	1	2	3	4
0	1	4	7	11	15
1	2	5	8	12	19
2	3	6	9	16	22
3	10	13	14	17	24
4	18	21	23	26	30

i

j target = 5

```
if ( matrix[i][j] > target)
    j--;
```

target = 20

(matrix[i][j] < target)

go down / right

(matrix[i][j] > target)

go left / up

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

target = 5

matrix[i][j] > 15 → go left

matrix[i][j] < 15 → go down

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

target = 13

ele > target go left

ele < target go down

target = 27

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

bahar → false

Thank you !!