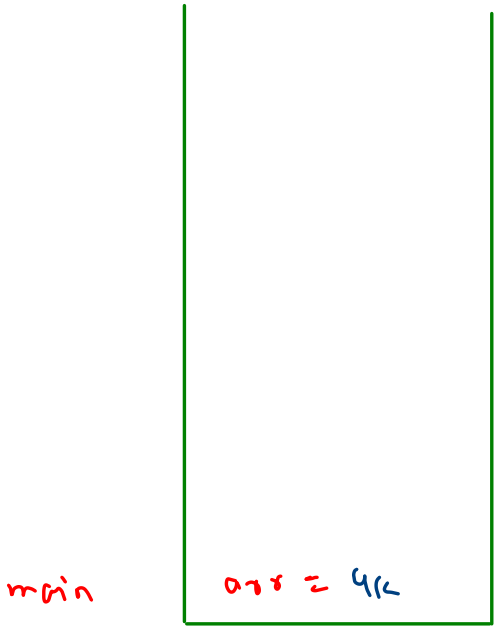
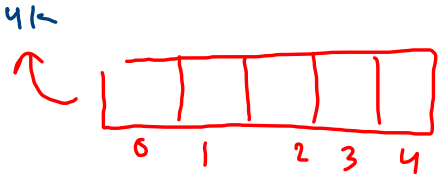
2d Arrays

Arrays

```
int [] arr = new int [5];
```





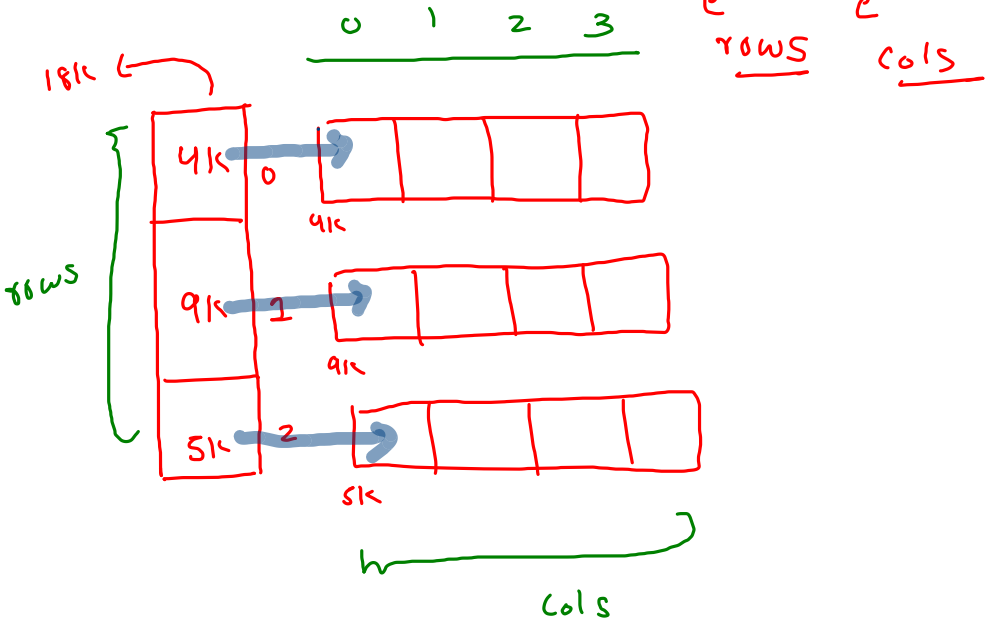
Variable

- primitive
  - (i) int — short, long, byte
  - (ii) double, float
  - (iii) char
  - (iv) boolean
- reference
  - arr

2d array

```
int [] arr = new int [5];
```

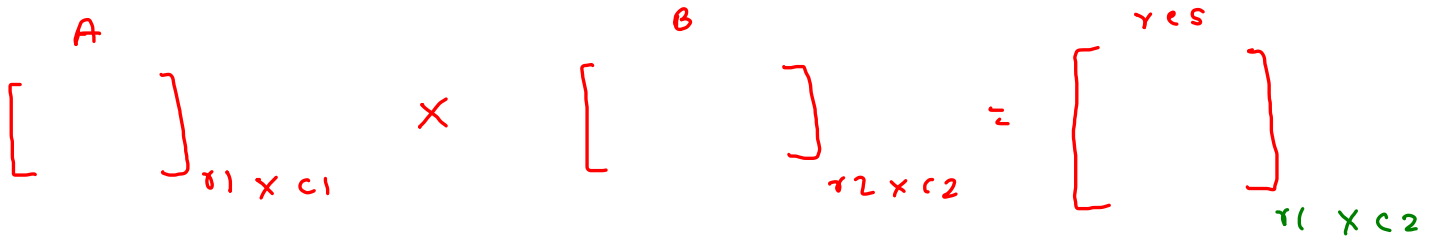
```
int[][] mat = new int [3] [4];
```



rows = mat.length

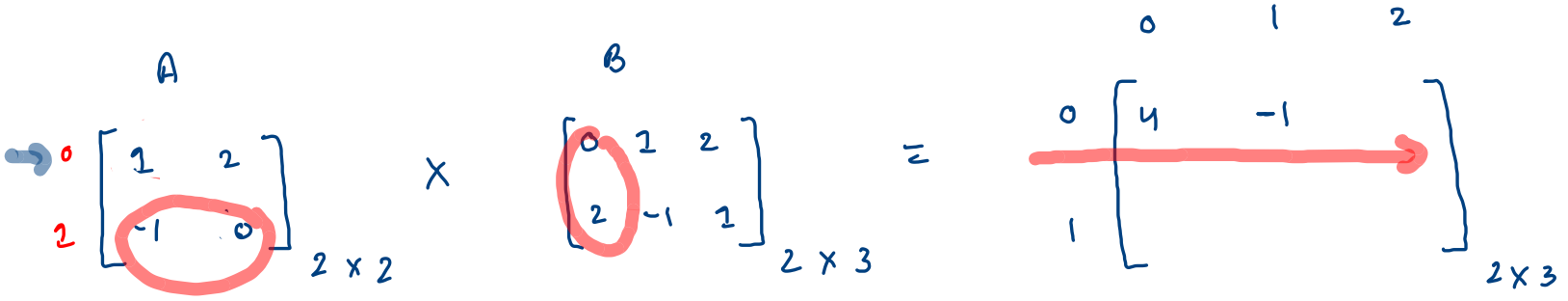
cols = mat[0].length

matrix mult



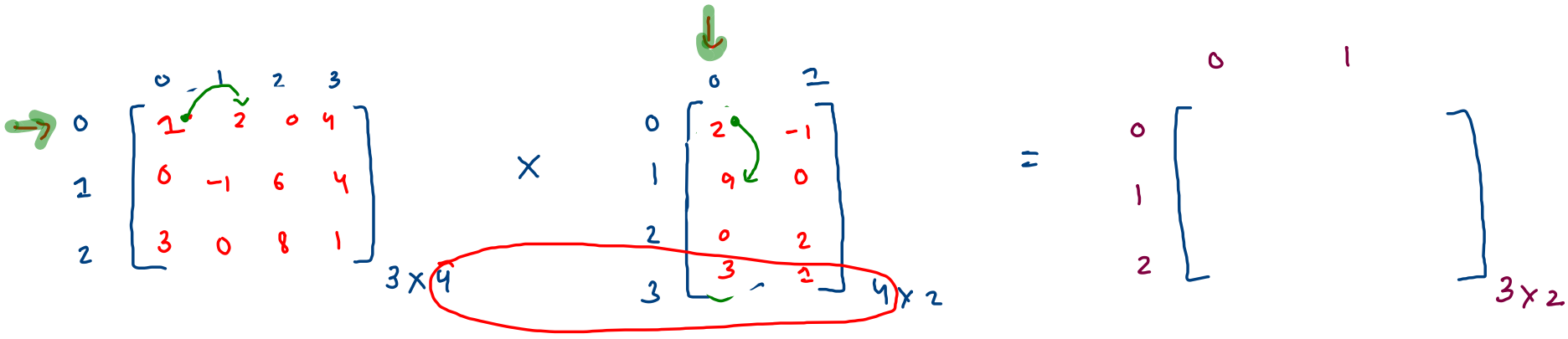
mult is possible : (c1 == r2)

res dimension : r1 , c2



res (i,j) -> A (ith row) , B (jth col)

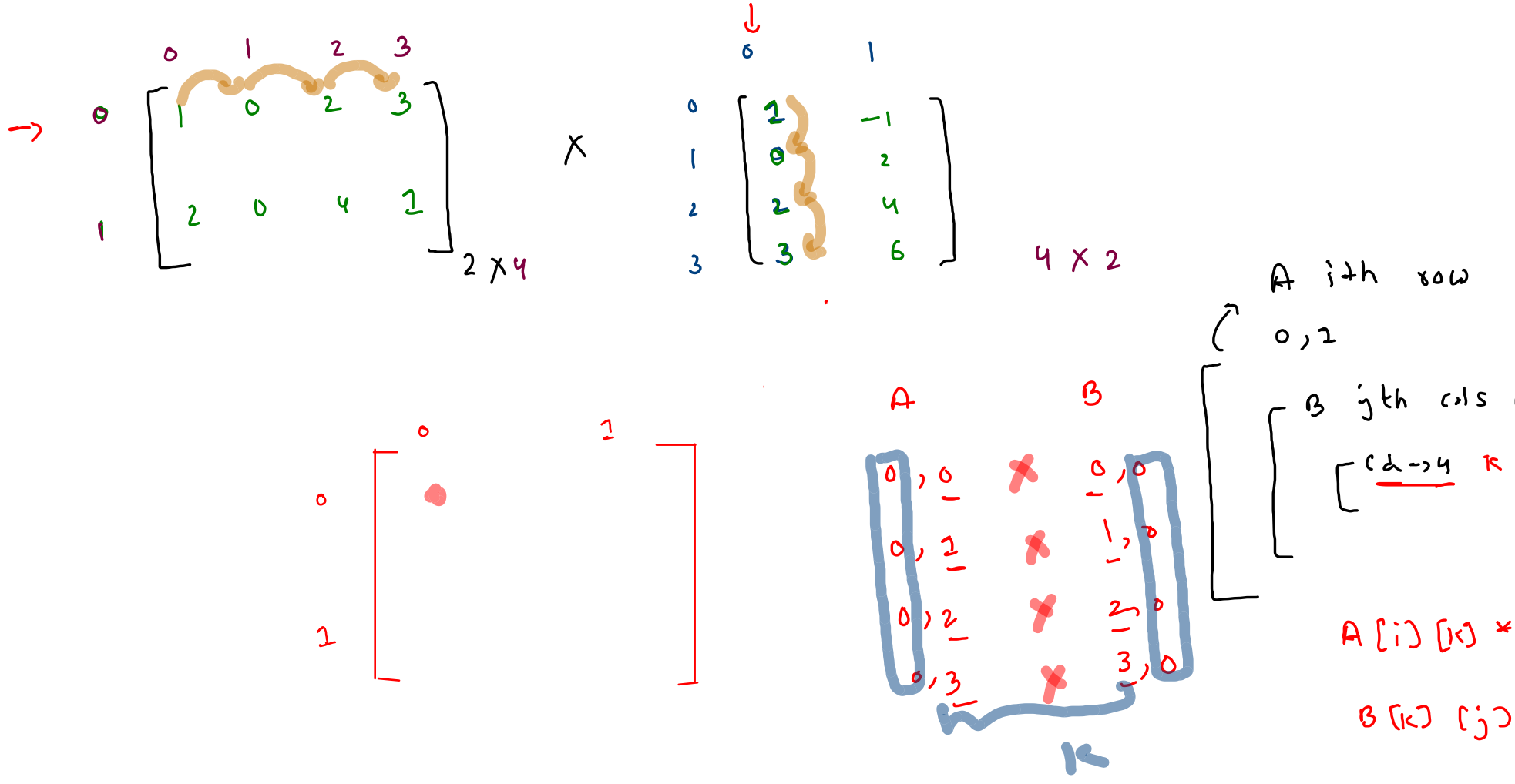
one on one mapping

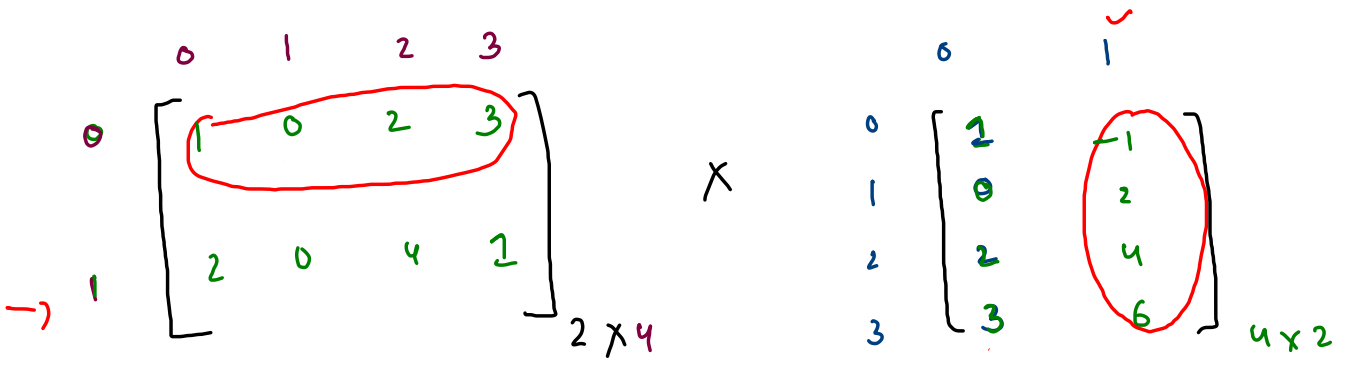


$i = 0, j = 0$

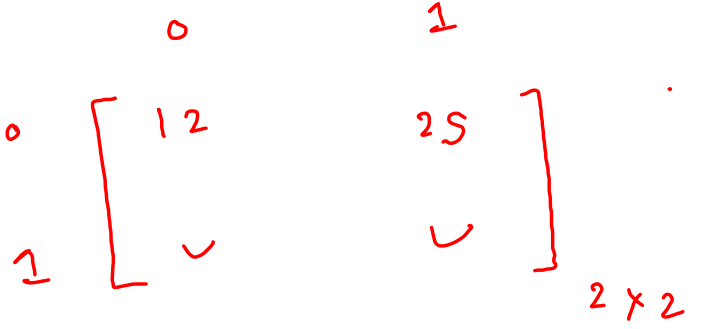
$res[0][0] = A[0][0] * B[0][0] + A[0][1] * B[1][0] +$

$A[0][2] * B[2][0] + A[0][3] * B[3][0]$


0 1 2 3
0
1
1 0 2 3
2 0 4 1
2 X 4
X
0 1
0
1
2
3
1 -1
0 2
2 4
3 6
4 X 2
A ith row
0,2
B jth cols 0,1
cl -> 4
A B
0,0 0,0
0,1 1,0
0,2 2,0
0,3 3,0
K
A [i] [k] *
B [k] [j]
0 1
0
1

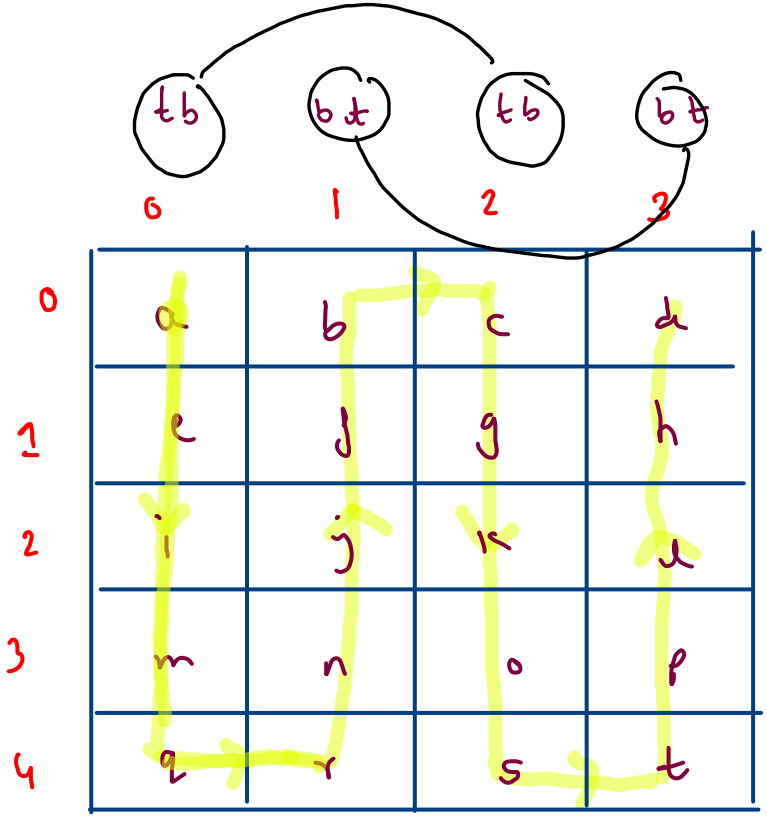


k = 0, 1, 2, 3




```
//to select ith row of A
for(int i=0; i < r1;i++) {
    //to select jth col of B
    for(int j=0; j < c2;j++) {
        //one on one mapping of A's ith row & B's jth col
        int val = 0;
        for(int k=0; k < c1;k++) {
            val += A[i][k] * B[k][j];
        }
        res[i][j] = val;
    }
}
```

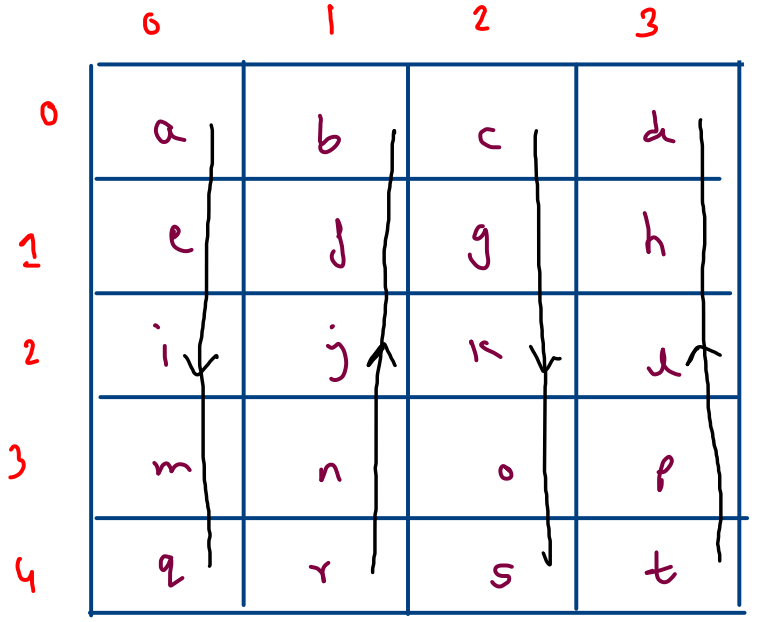
val = -1 + 0 + 8 + 18 = 25



However, a certain visitor decides to travel a different path as follows:
1. He first travels southwards till no further south places are available.
2. He then moves only 1 place eastwards.
3. He starts to move again towards north till any further north moves are available.
This continues till all the places are covered.

a e i m q r n j f b

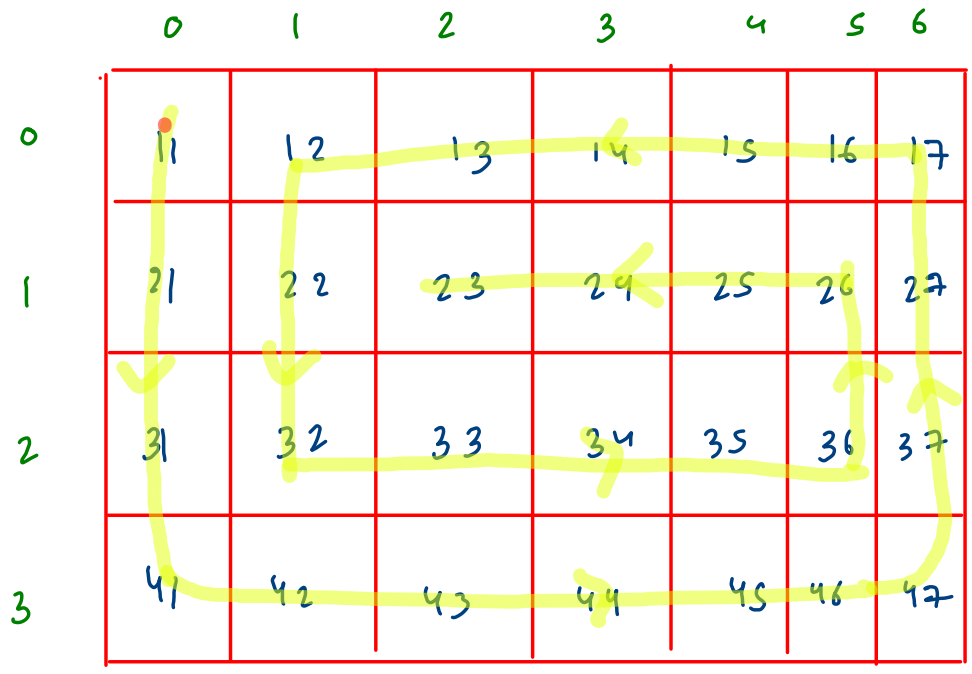
c g k o s t p l h d



even cols → top to bottom

odd cols → bottom to top

a e i m q r n j f b
c g k o s t p l h d

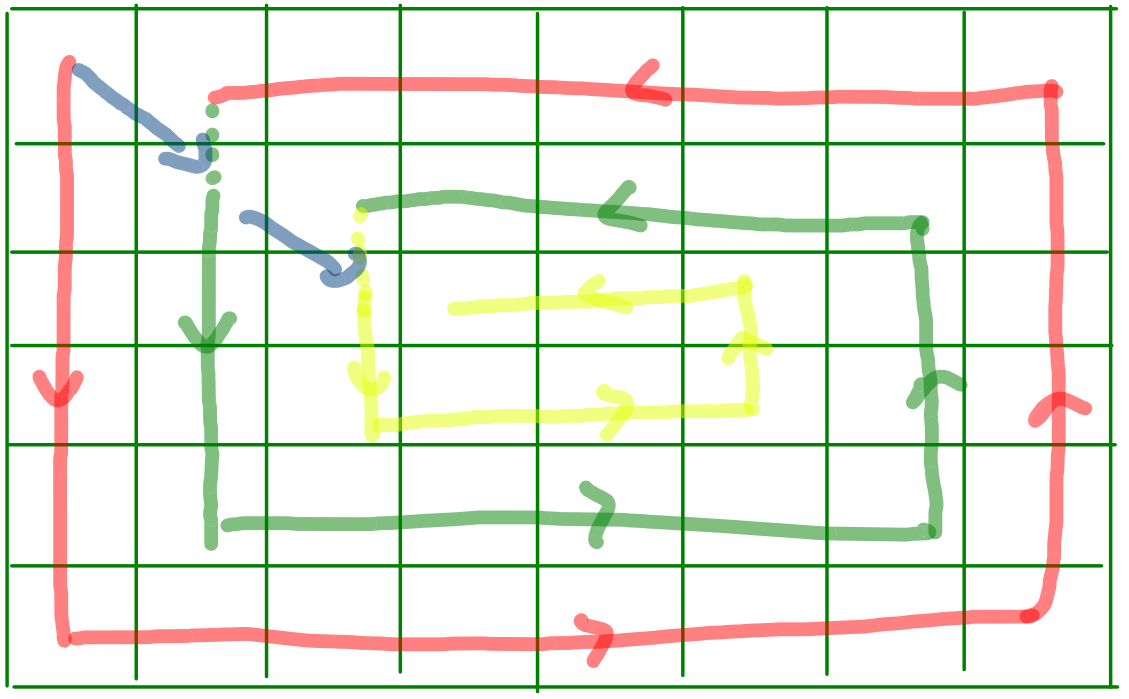


11 21 31 41 42 43 44

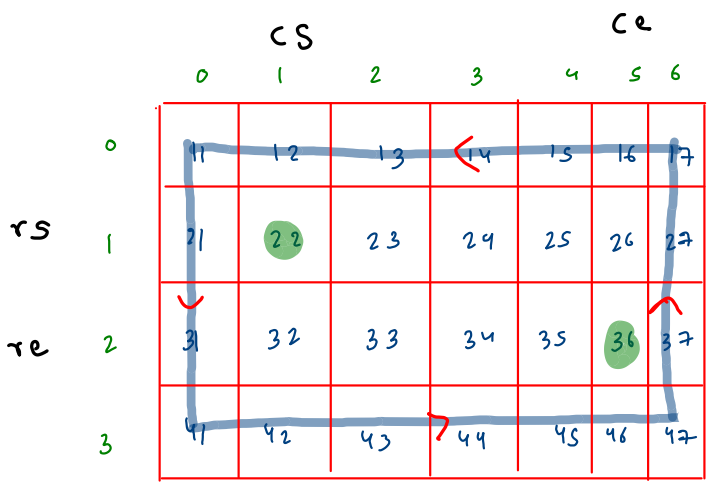
45 46 47 37 27 17 16

15 14 13 12 22 32 33

34 35 36 26 25 24 23



spiral is made of shells.



shell : 4 walls

left wall, bottom wall, right wall,

top wall.

left wall :

```
for( int r = rs ; r <= re ; r++ ) {
    syso ( mat [r] [cs]);
}
cs++;
```

bottom wall ;

```
for( int c = cs ; c <= ce ; c++ ) {
    syso (mat [re] [c] );
}
re--;
```

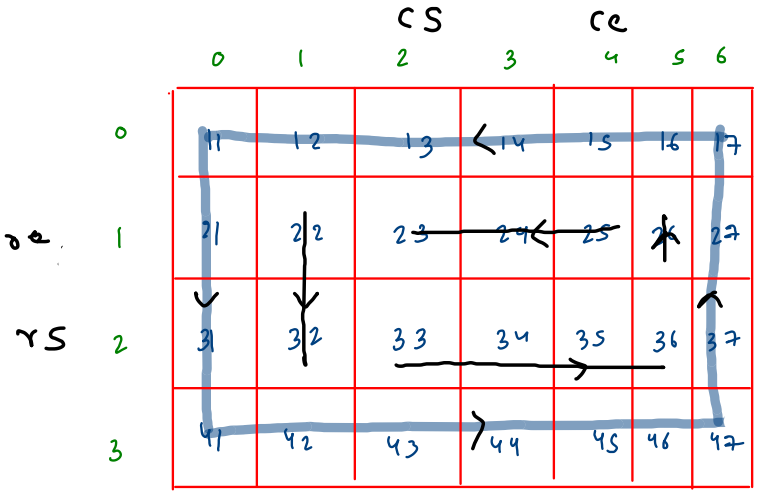
right wall

```
for( int r = re ; r >= rs ; r--) {
    syso ( mat [r] [ce] );
}
ce--;
```

top wall

```
for (int c = ce ; c >= cs ; c--){
    syso ( mat [rs] [c ] );
}
rs++;
```

```
while ( psf < tne ) {

  left wall :

  for( int r = rs ; r <= re ; r++ ) {
      syso ( mat [r] [cs]); psf++;
  }
  cs++;

  bottom wall ;
  for( int c = cs ; c <= ce ; c++ ) {
       syso (mat [re] [c] ); psf++;
  }
  re--;

  right wall
  for(int r = re ; r >= rs ; r--){
     syso ( mat [r] [ce]); psf++;
  }
  ce--;

   top wall
  for (int c = ce ; c >= cs ; c--){
       syso ( mat [rs] [c] ); psf++;
  }
  rs++;

}
```



tne = 28

11 21 31 41 42 43 44 45

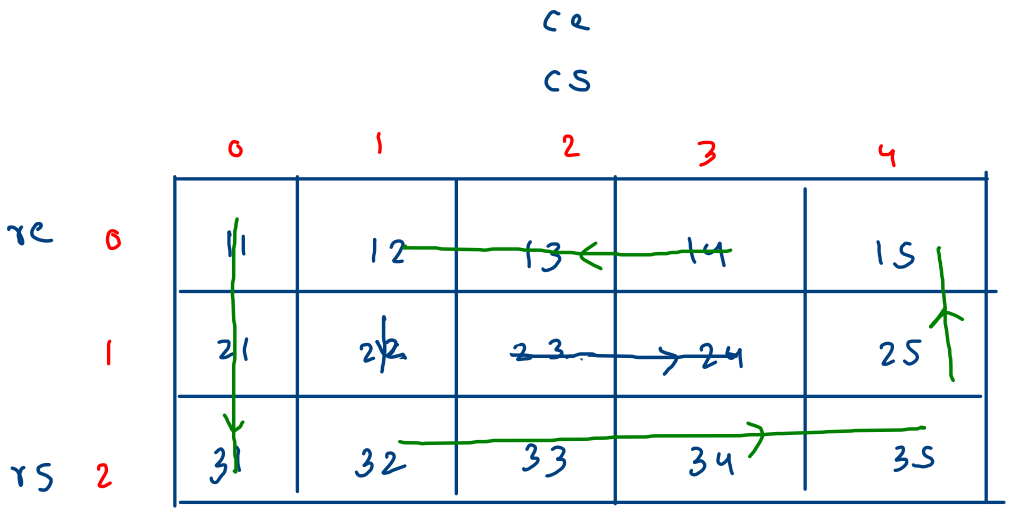
psf = 28

46 47 37 27 17 16 15 14

13 12 22 32 33 34 35 36

26 25 24 23

```
while(psf < tne) {
    //left wall
    for(int r=rs; r <= re;r++) {
        System.out.println(mat[r][cs]);
        psf++;
    }
    cs++;

    //bottom wall
    for(int c=cs; c <= ce;c++) {
        System.out.println(mat[re][c]);
        psf++;
    }
    re--;

    //right wall
    for(int r=re; r >= rs;r--) {
        System.out.println(mat[r][ce]);
        psf++;
    }
    ce--;

    //top wall
    for(int c=ce;c >= cs;c--) {
        System.out.println(mat[rs][c]);
        psf++;
    }
    rs++;
}
```



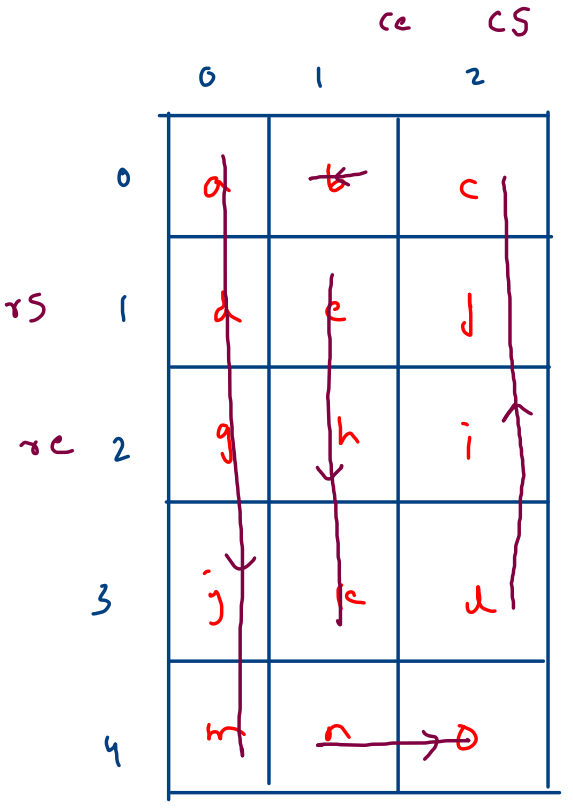
tne = 15

11 21 31 32 33 34 35 25 15

14 13 12 22 23 24 23

psf = ~~12~~ ~~15~~ 16

```java
while(psf < tne) {
    //left wall
    for(int r=rs; r <= re && psf < tne;r++) {
        System.out.println(mat[r][cs]);
        psf++;
    }
    cs++;

    //bottom wall
    for(int c=cs; c <= ce && psf < tne;c++) {
        System.out.println(mat[re][c]);
        psf++;
    }
    re--;

    //right wall
    for(int r=re; r >= rs && psf < tne;r--) {
        System.out.println(mat[r][ce]);
        psf++;
    }
    ce--;


    //top wall
    for(int c=ce;c >= cs && psf < tne;c--) {
        System.out.println(mat[rs][c]);
        psf++;
    }
    rs++;
```



tne = 15

psf = 0 1 2 ... 15

if

```
if ( c1 ) {
    // w1
}
if ( c2 ) {
    // w2
}
else {
    // w3
}
```

c1 -> T

c2 -> T

w1 ✗

w2 ✓

else if

```
if ( c1 ) {
    // w1
}
else if ( c2 ) {
    // w2
}
else {
    // w3
}
```

w1 ✗