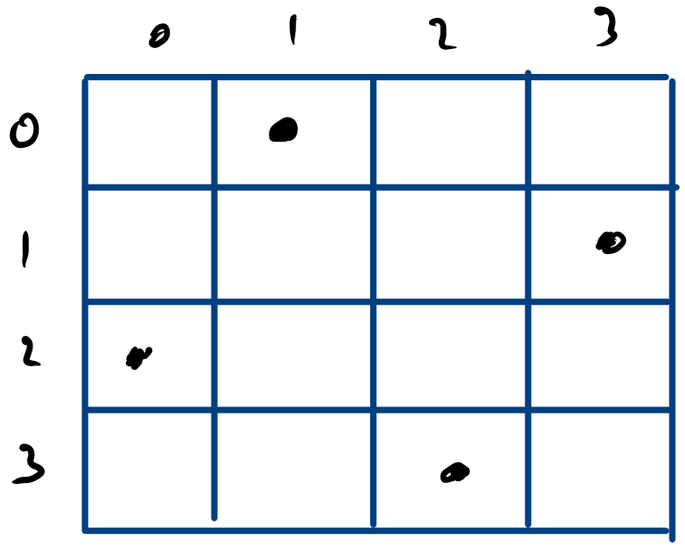
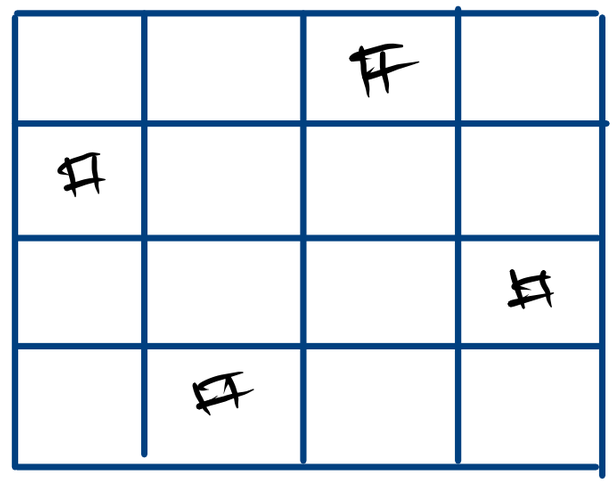
n = 4

4 Q



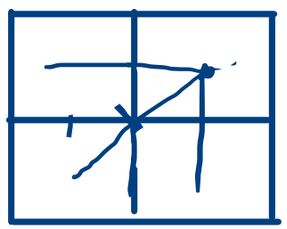


0-1, 1-3, 2-0, 3-2, .

0-2, 1-0, 2-3, 3-1, .

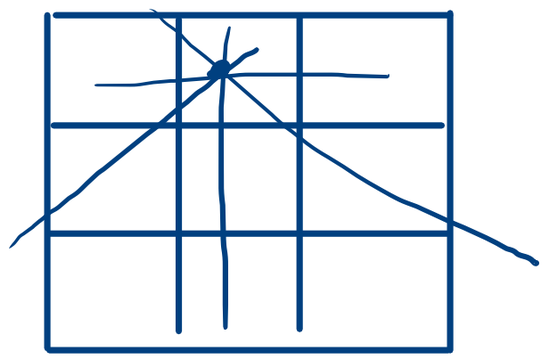
n=2

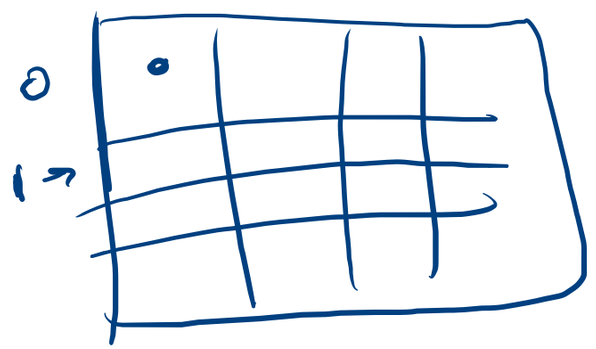
22



—

n=3





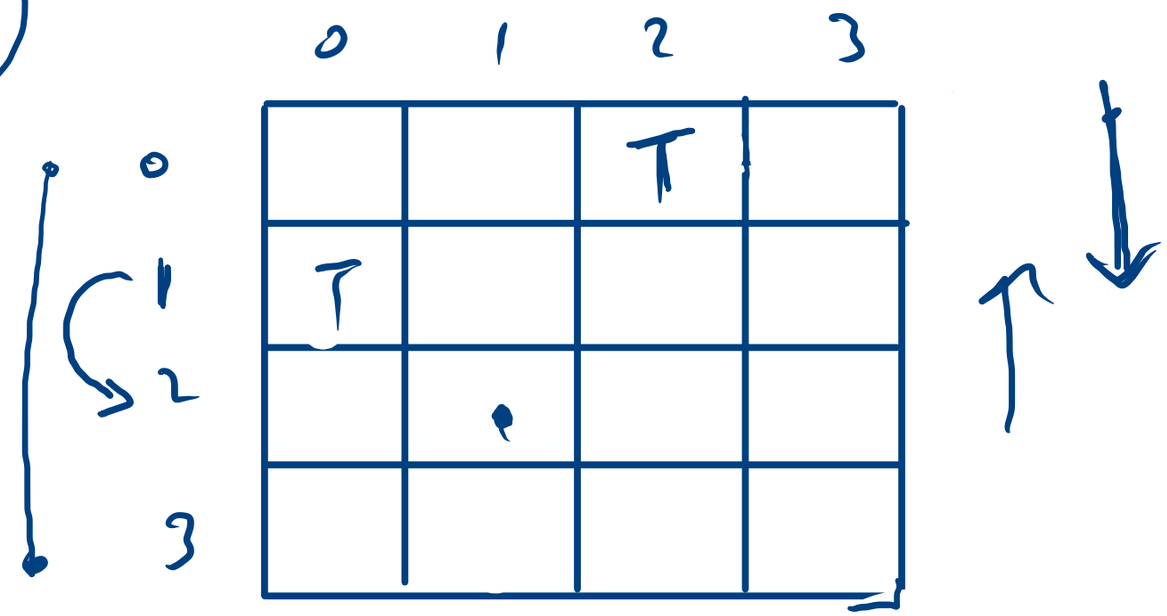
✓
✓

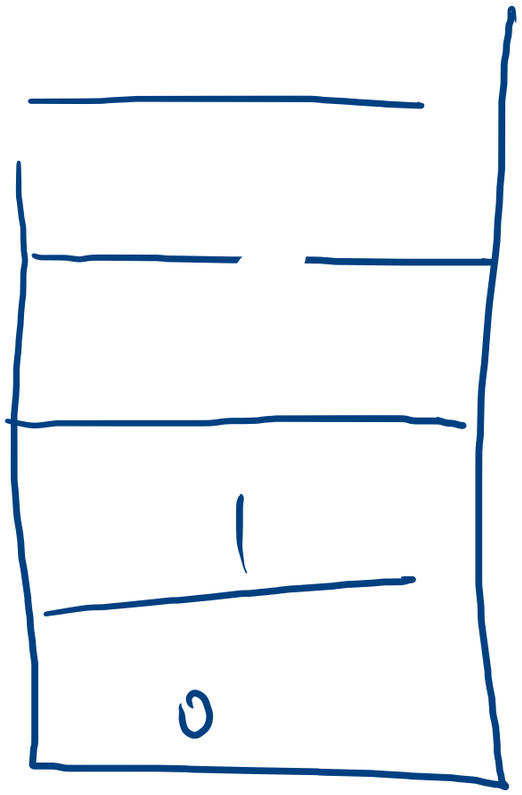
n

row -> 1

blank -> false

h=4





for(j 0 .... h-1)

row, j

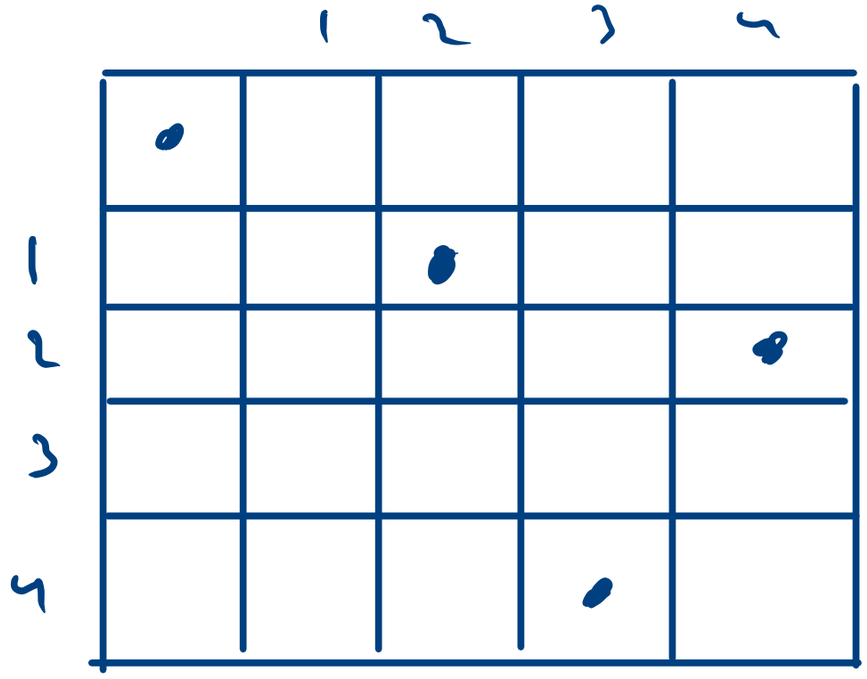
if(safe)

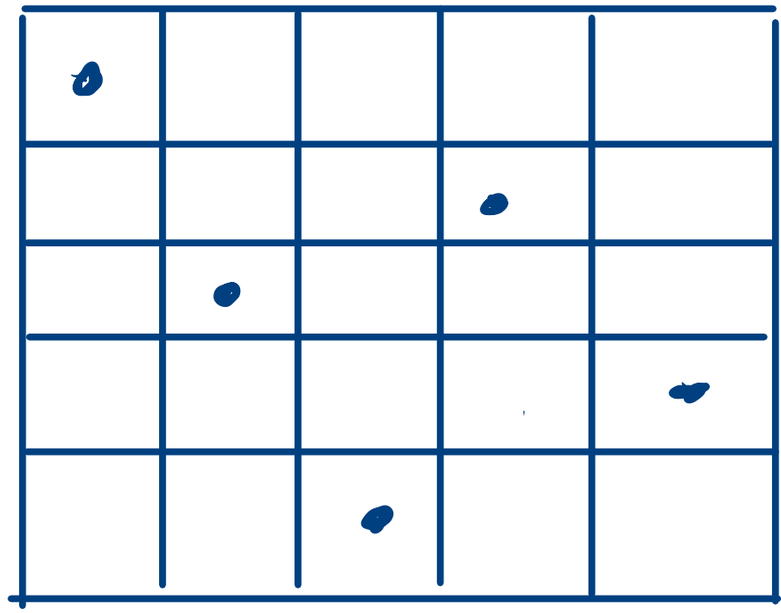
place Q

finish(row+1)

remove Q

right





0-0, 1-2, 2-4, 3-1, 4-3, .

0-0, 1-3, 2-1, 3-4, 4-2, .

5 2 0

5x5

	0	1	2	3	4
0	25	2	13	8	19
1	12	7	18	3	14
2	1	24	15	20	9
3	6	11	22	17	4
4	23	16	5	10	21

```
25 2 13 8 23
12 7 24 3 14
1 18 15 22 9
6 11 20 17 4
19 16 5 10 21
```

h=5

5x5

```
19 2 13 8 21
12 7 20 3 14
1 18 15 22 9
6 11 24 17 4
25 16 5 10 23
```

```
25 2 13 8 19
12 7 18 3 14
1 24 15 20 9
6 11 22 17 4
23 16 5 10 21
```

0,0

21

4,0

3,2

4,4

2,3

0,4

1,2

3,3

25

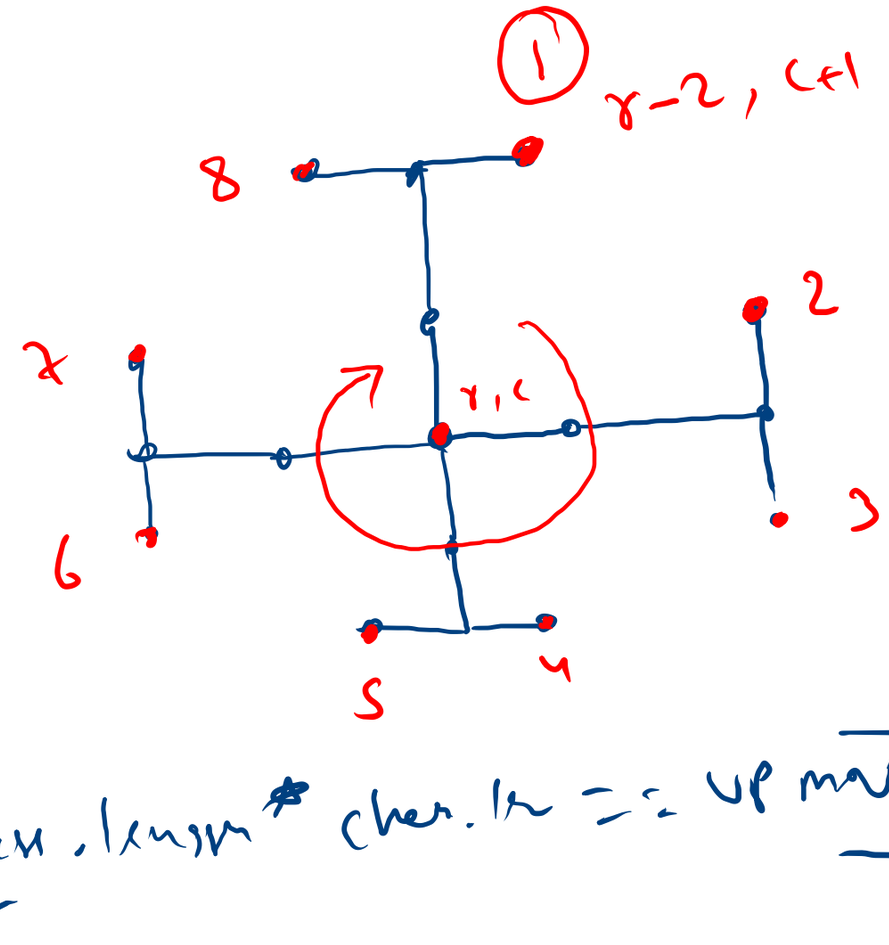
18

0,0 25

2,0

2	3	6
4	2	5
3	4	4
1	3	3
0	1	2
2	0	1
r	c	up

	0	1	2	3	4
0	25	2			
1				3	
2	1			6	
3					4
4			5		



allocate & Init

[chess.length * chess.length == up.max]

```java
if(r<0 || c< 0 || r>=chess.length ||
c >= chess.length ||
chess[r][c] != 0)return;

if(chess.length*chess.length == upcomingMove){
    chess[r][c] = upcomingMove;
    displayBoard(chess);
    chess[r][c] = 0;
    return;
}

chess[r][c] = upcomingMove;
printKnightsTour(chess, r-2, c+1, upcomingMove+1);
printKnightsTour(chess, r-1, c+2, upcomingMove+1);//
printKnightsTour(chess, r+1, c+2, upcomingMove+1);
printKnightsTour(chess, r+2, c+1, upcomingMove+1);//
printKnightsTour(chess, r+2, c-1, upcomingMove+1);//
printKnightsTour(chess, r+1, c-2, upcomingMove+1);
printKnightsTour(chess, r-1, c-2, upcomingMove+1);
printKnightsTour(chess, r-2, c-1, upcomingMove+1);

chess[r][c] = 0;
```

5×5 = 25

```
25  2 13  8 23
12  7 24  3 14
 1 18 15 22  9
 6 11 20 17  4
19 16  5 10 21
```

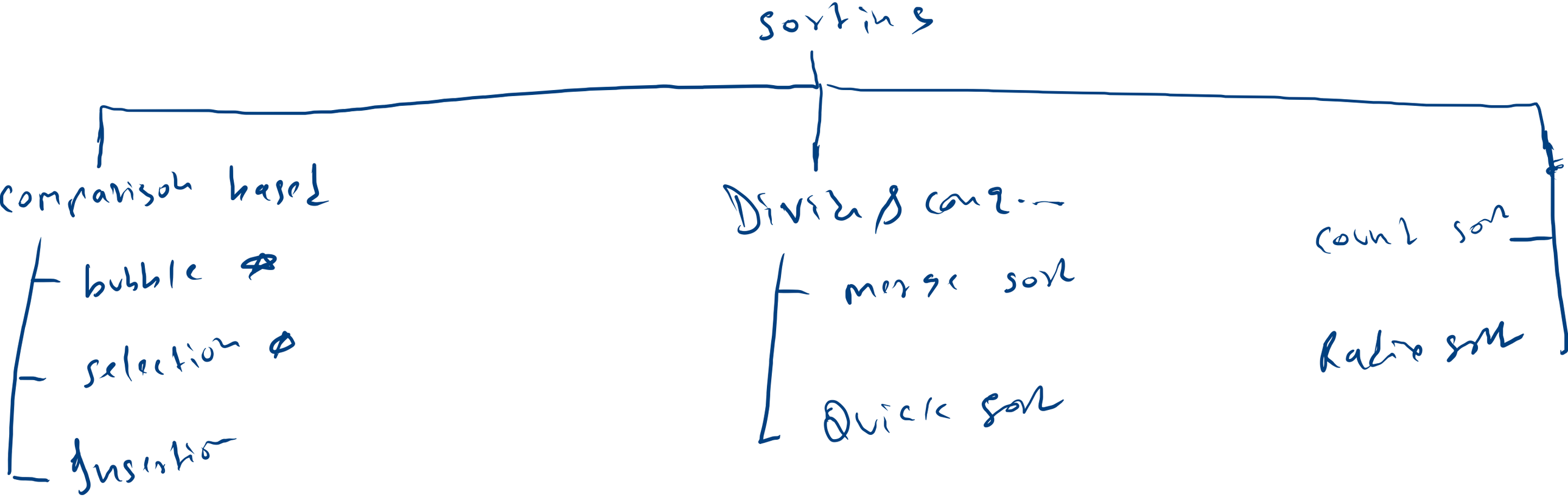
a b c d e f g h

(1) (2)

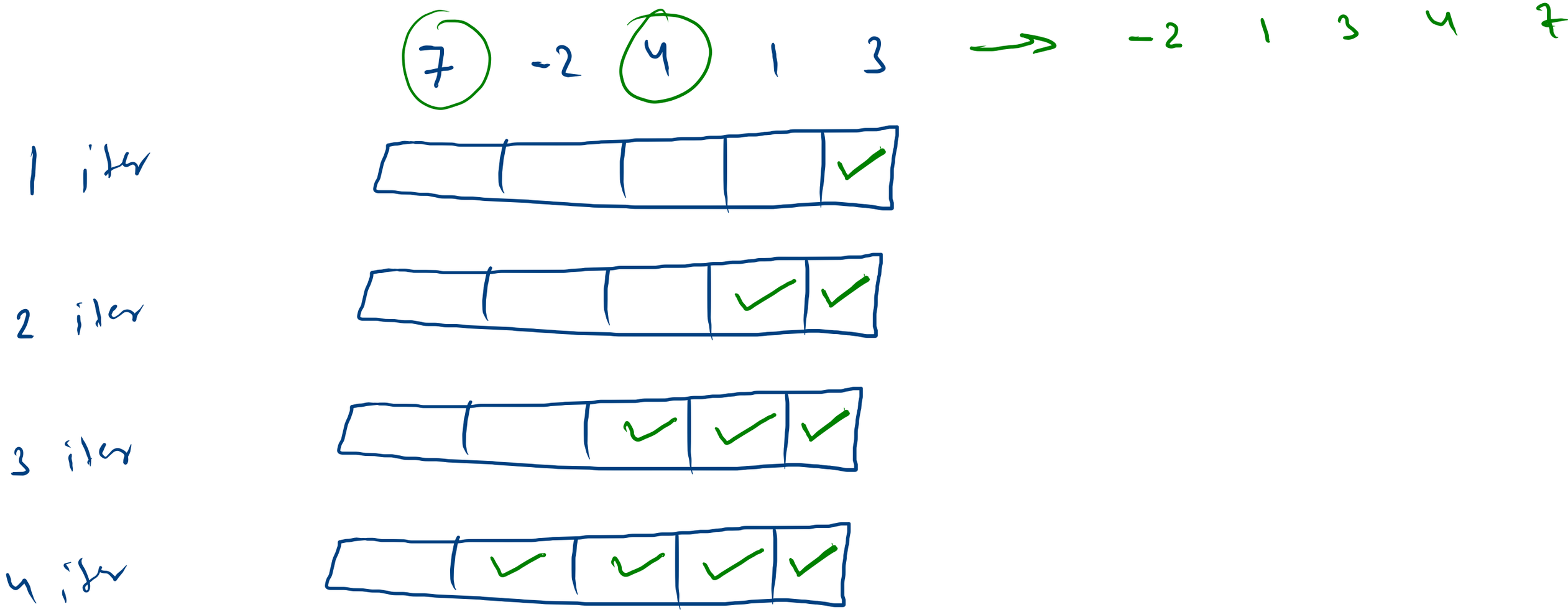
	0	1	2	3	4
0	19	2	13	8	21
1	12	7	20	3	14
2	1	18	15	22	9
3	6	11		17	4
4		16	5	10	23

4	4	23
2	3	22
0	4	21
1	2	20
0	0	19
2	1	18
3	3	17

c			6
	4	2	5
	3	4	4
	1	3	3
	0	1	2
	2	0	1
	r	<	UP


Sorting
comparison based
bubble
selection
Insertion
Divide conq...
merge sort
Quick sort
Count sort
Radix sort

Bubble Sort



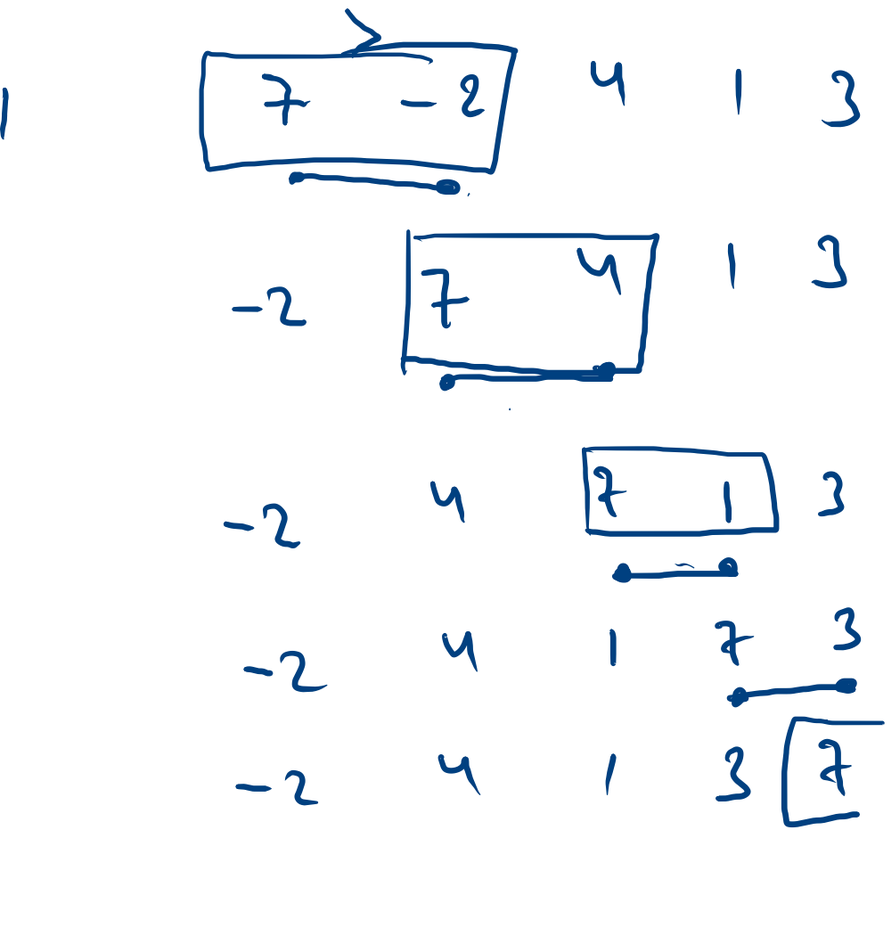
n-1 iterations

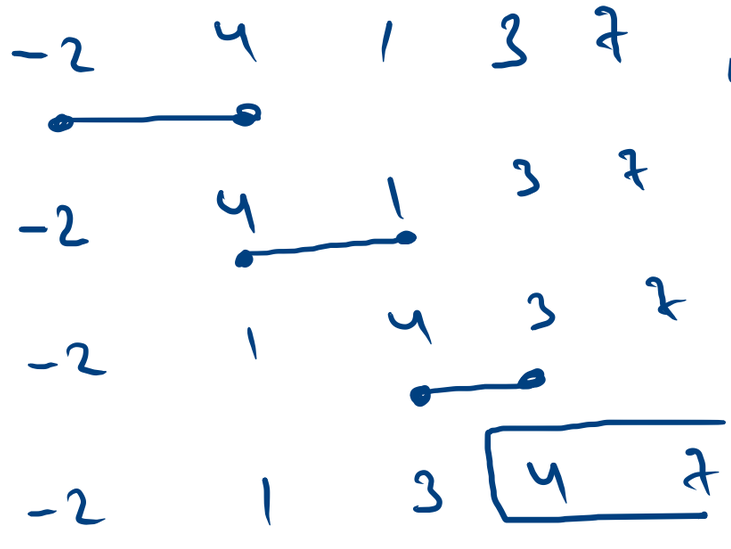
i = 4, 3, 2, 1

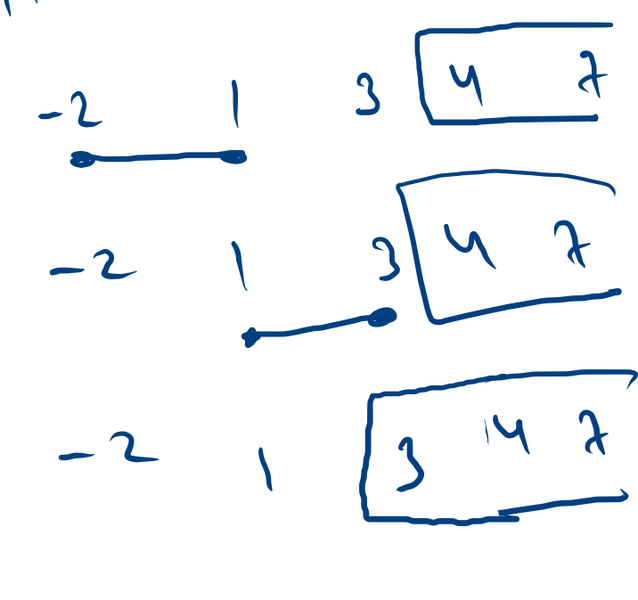
7 -2 4 1 3

#2

#3





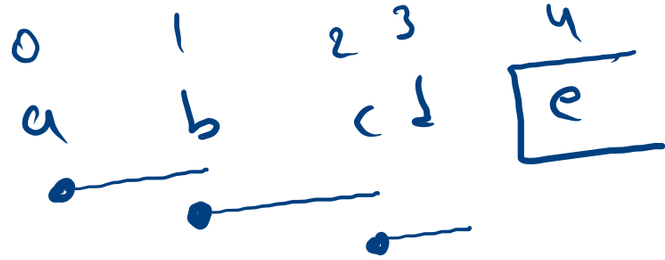


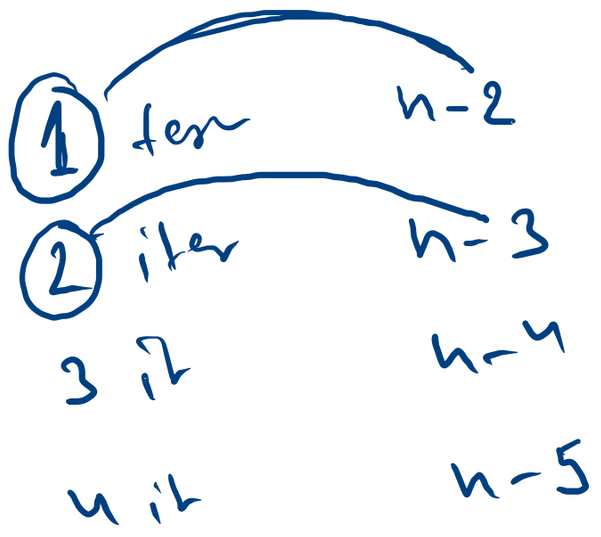
n

```
for( int iter = 1; iter ≤ n-1; iter ++) {
    for( int j = 0; j ≤ n - (iter+1); j++) {

    }
}
```

n = 5





$n^2 + n$

n^2

```java
int n = arr.length;

for(int iter = 1; iter <= n-1; iter++){
    for(int j = 0; j<= n - (iter+1); j++){
        if(isSmaller(arr, j+1, j)){
            swap(arr, j+1, j);
        }
    }
}
```

n-1

iter

iter	
1	n-1
2	n-2
3	n-3
4	n-4
⋮	⋮
n-1	1

$(n-1) + (n-2) + (n-3) \cdots\cdots 1$

$= \frac{n \times (n+1)}{2} - n$

$= \frac{n^2 + n}{2} - n$

$= \frac{n^2}{2} + \frac{n}{2} - n$

$= \frac{1}{2} \times n^2$

$= O(n^2)$

order

$n^2 + n$ → order → (n^2)

$2n^2 + 3$ → order → (n^2)