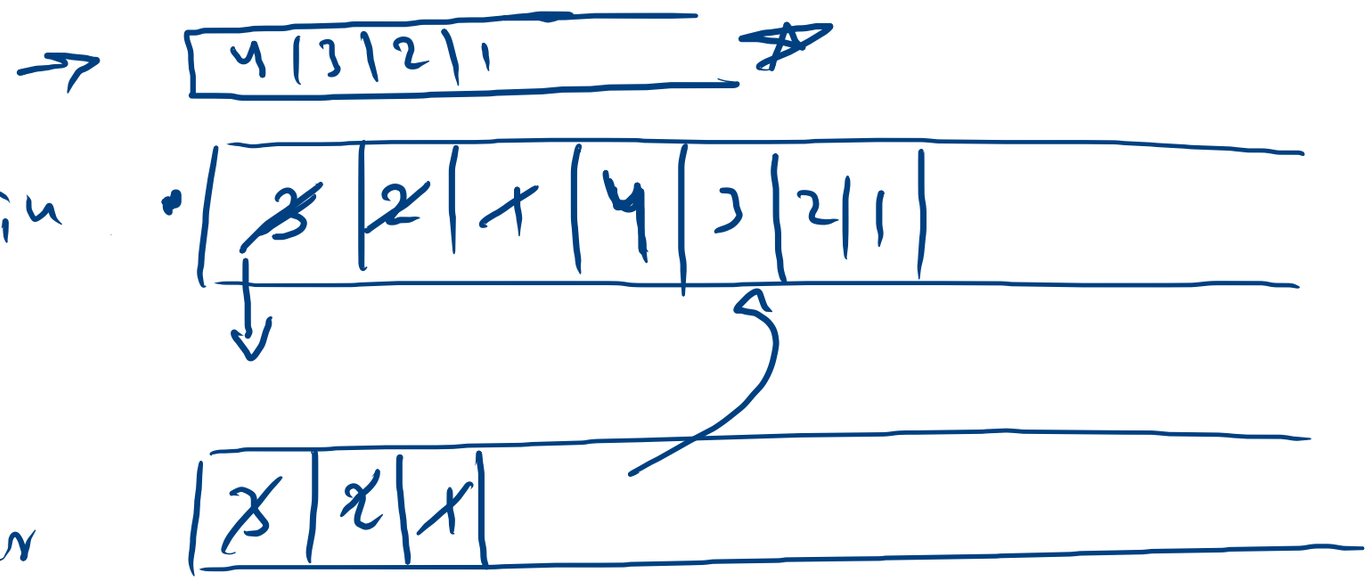
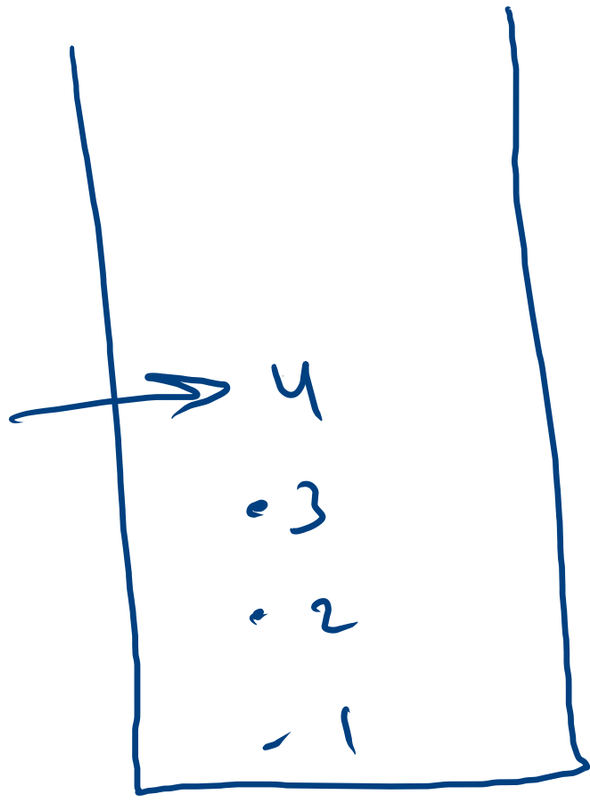




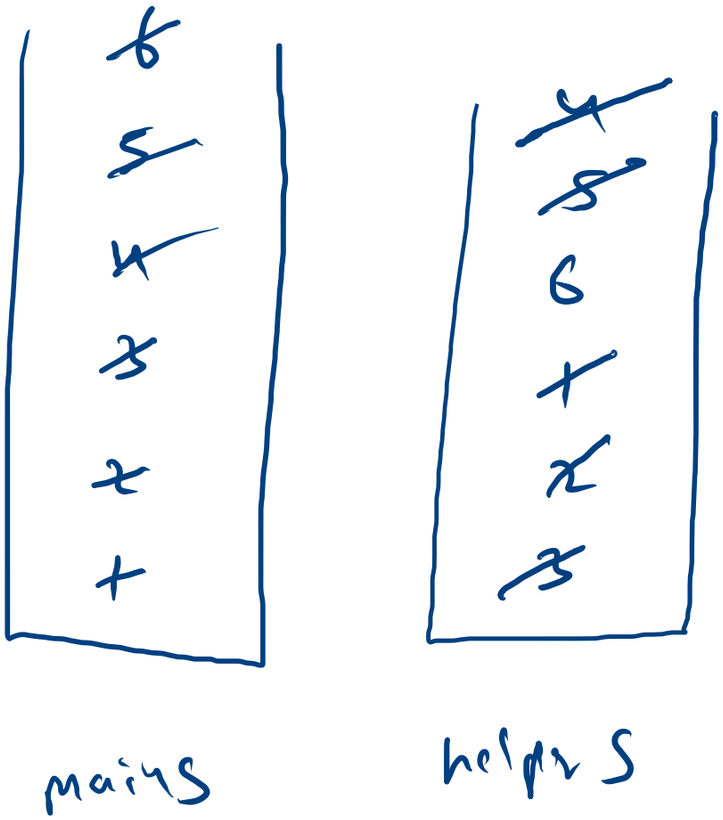
pop efficient

pop main.remove

push main -> helper

main.add(val)

helper -> main

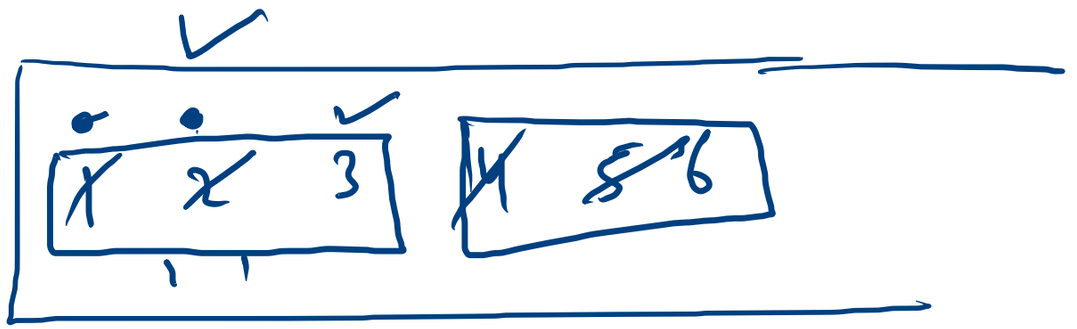


mains

helper S

val → 1

1
2
3
4
5

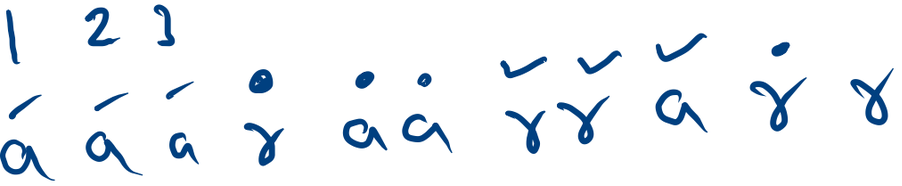


add → mains.push

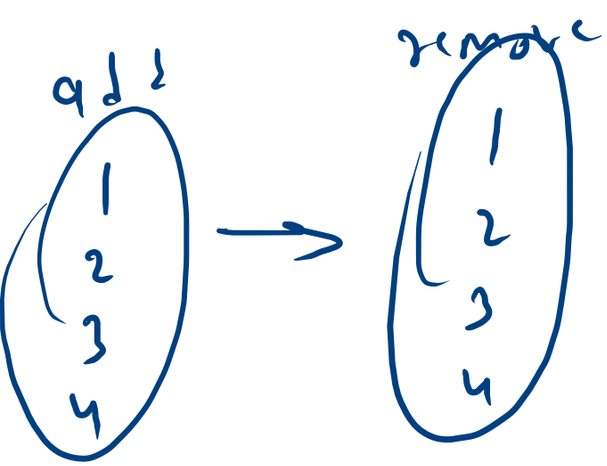
remove → helper.size > 0
helper.remove return

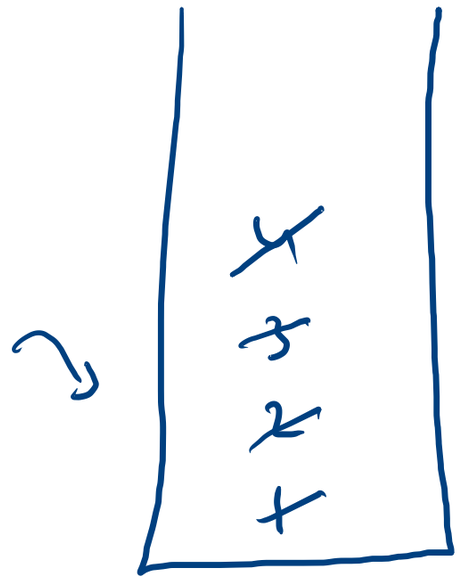
main → helper
helper.remove return

1 2 3

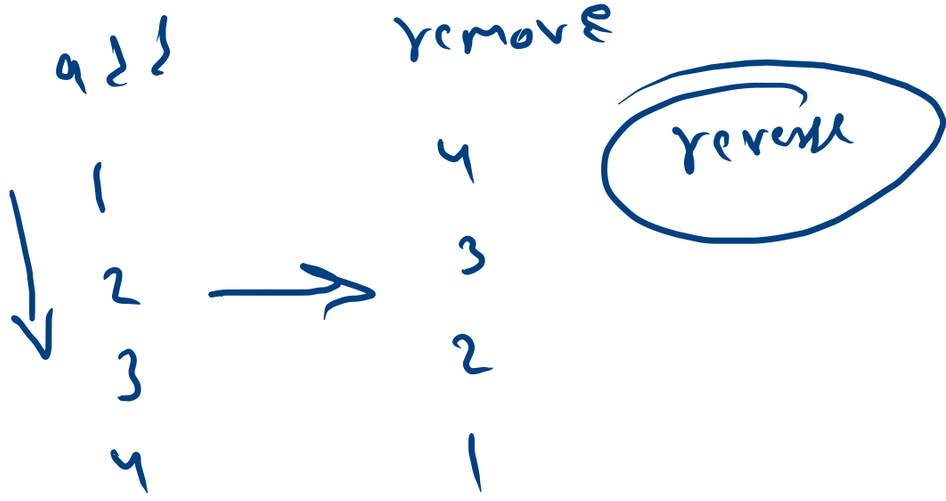


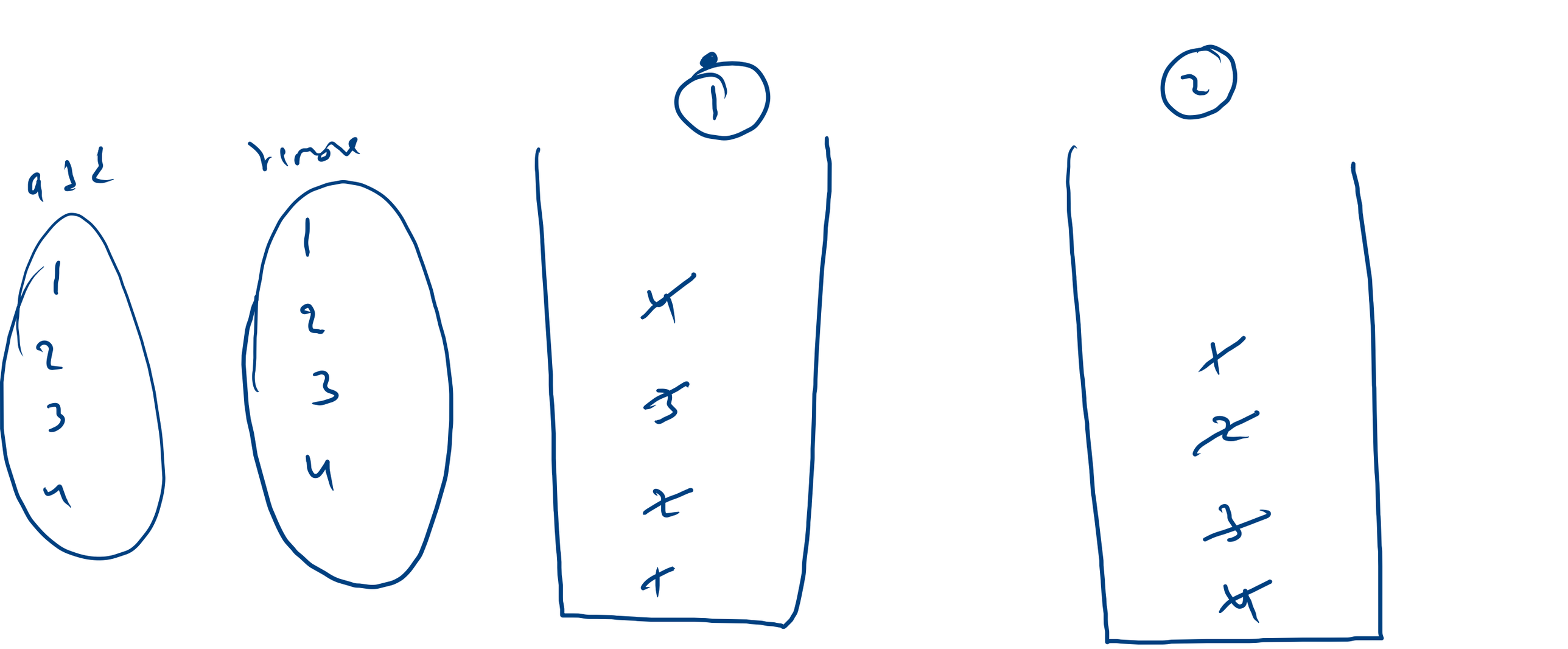
4	2	3	4





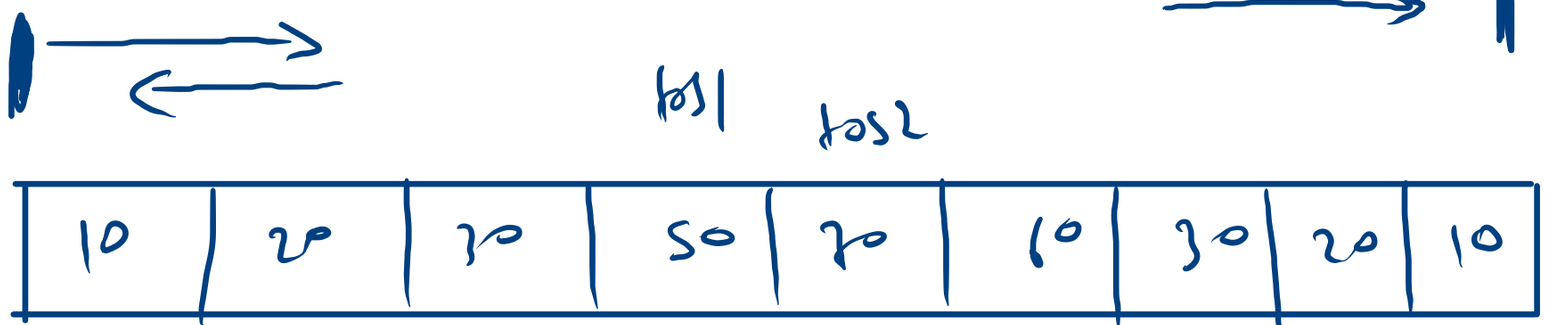
$((1234)^1)^1$




q 1 2
1
2
3
4
remove
1
2
3
4
1
2

9

50
~~40~~
30
20
10



70
60
30
20
10

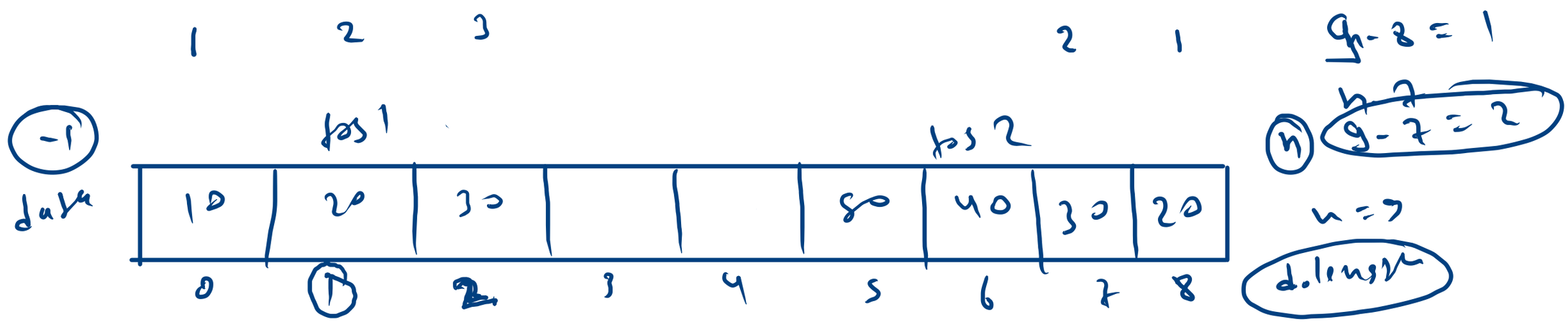
S1

data

top1

top2

S2



push1

pop1

top1

isize1

tos1++ data[tos1]=val

tos1+1

10

S1

push2

pop2

top2

size2

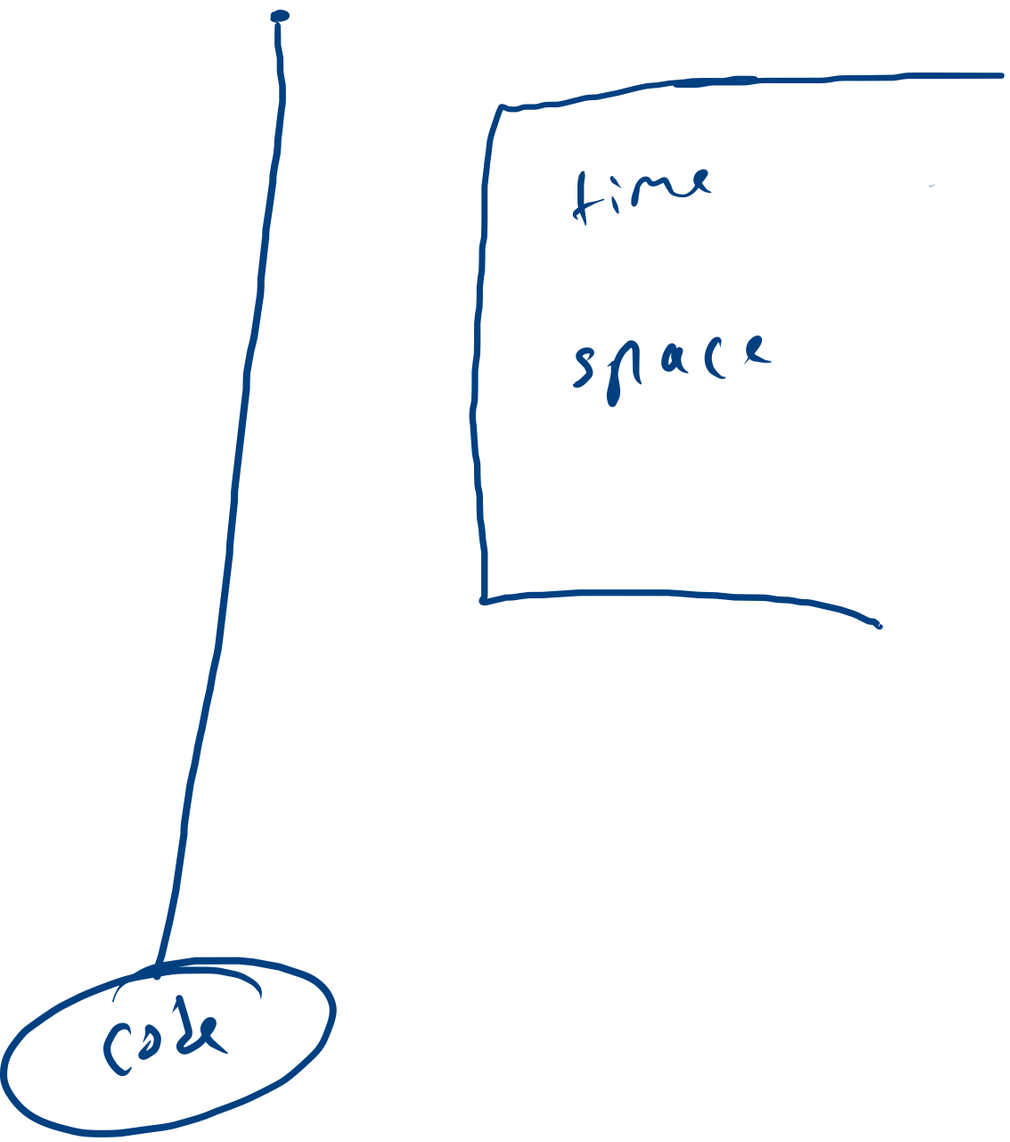
d.len-tos2

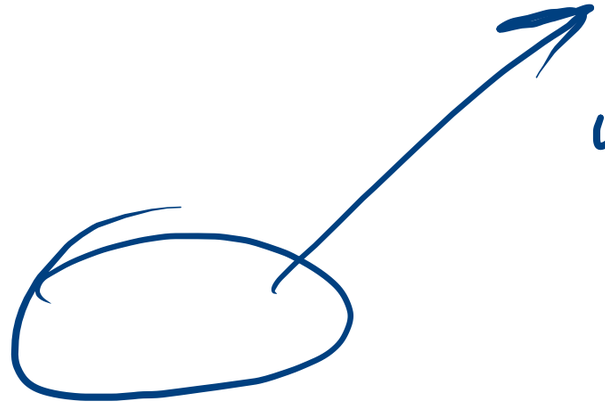
30

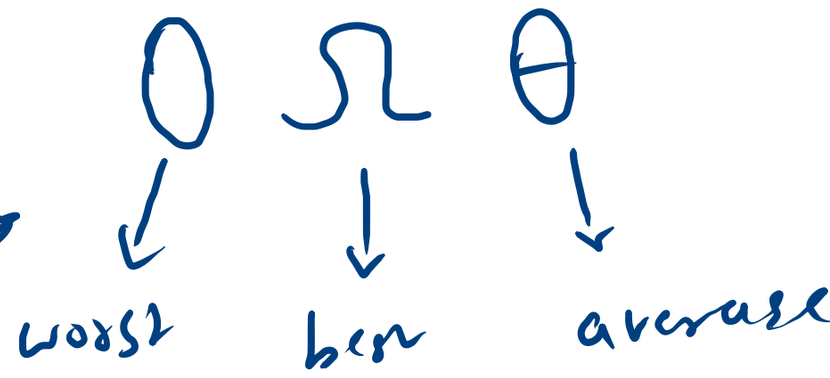
20

S2

A1 A2 A3 A4

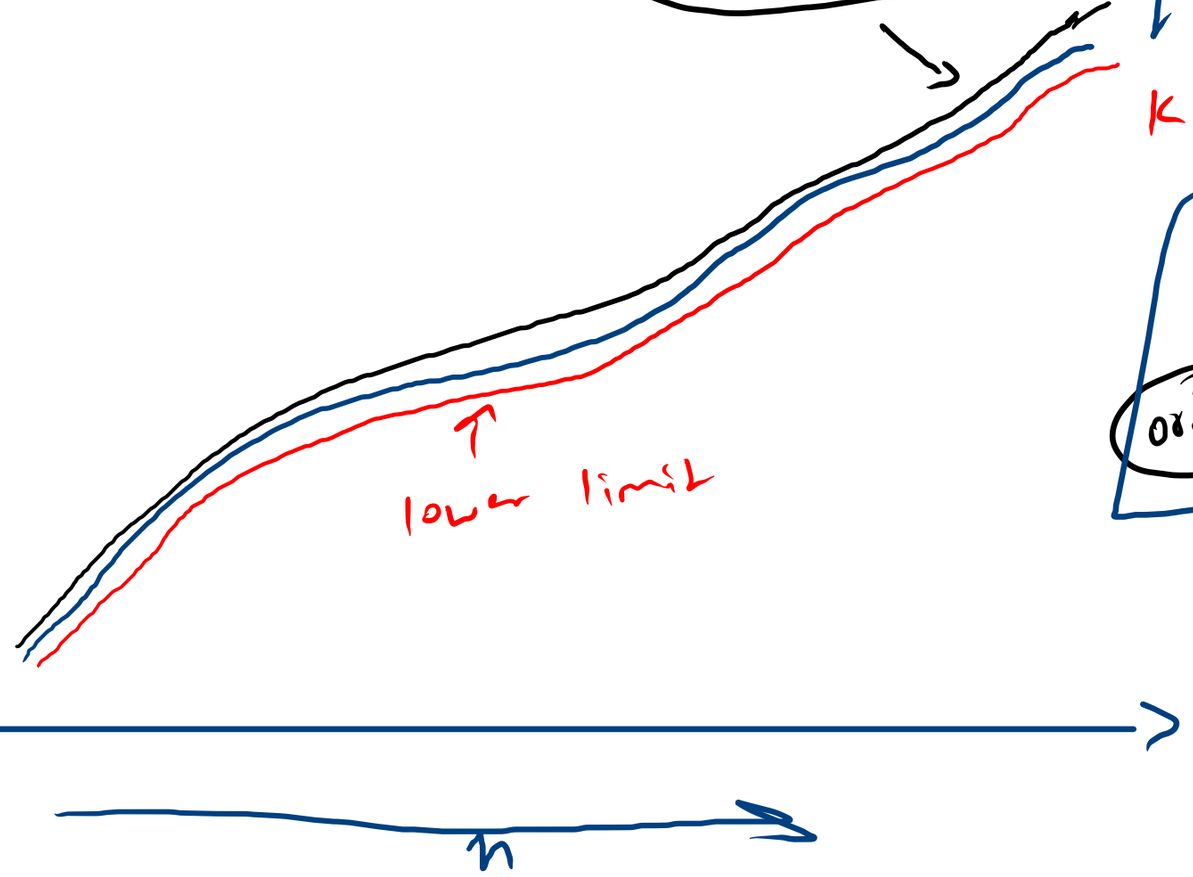






0

T(n)



parra

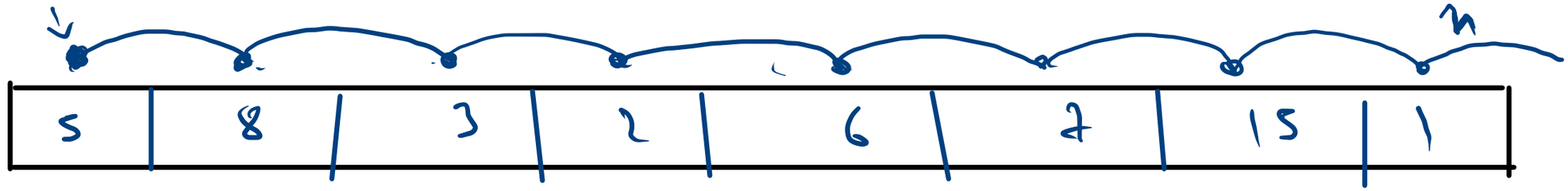
heaven up.. limit

$f(n) \leq c.g(n)$

order $\Rightarrow O(g(n))$

$f(n) \geq K.M(n)$

$\Omega(M(n))$



Linear Search

find 2 4

find (5) 1

find (100) n

$f(n)$

$f(n) \geq 1$

best → $\Omega(1)$

$f(n) \leq n \cdot 1$

worst case $O(n)$

```
for(int i=0;i<n;i++){
    ≡ constum  c
}
```

i	
0	c
1	c
2	c
3	c
⋮	
n-2	c
n-1	c
n	-

$g(n) =$ n . c

$\Rightarrow O(n)$

```
for( int i=0;i<n;i++){
    for(int k=0;k<n;k++){
        } constant c
    }

    for( int j=1;j<=10;j++){
        } constant k
    }
}
```

$i = 0$ → k 0 .... n → $n.c + 10.k$

$i = 1$ → k 0 .... n → $n.c + 10.k$

$i = 2$ → k 0 .... n

⋮

$i = n-1$ → k 0 .... n → $n.c + 10.k$

$n.c$

$n(n.c + 10.k)$

$n^2.c + n\,10.k$

$n^2 + n = f(n)$

$n^2 + n^2 = g(n)$

$2n^2$

$O(n^2) \in$ $\in$

n

$n * (n.c) \rightarrow n^2.c$

$O(n^2)$ $\in$

$f(n) \leq g(n)$

$f(n) \leq g(n)$

n variable
k variable

n > k
n < k
n, k

```
for( i=1; i<=n; i += k){
    for( j = 1; j<=k; j++){
        = } c
    }
}
```

$(m+1)\,k \cdot c \rightarrow \left(\frac{n}{k}+1\right) k \cdot c$

≈
0

$mk+1 \le n$	$\frac{nk}{k} + k \cdot c$
$mk \le n-1$	$n + k$
$m \le \frac{n-1}{k}$	$n + n$
$m = \frac{n}{k}$	$2n$
	$O(n)\ O(k)\ O(n+k)$

i		
1	1	k . c
1	1 k+1	k . c
1	2 k+1	k . c
1	3 k+1	k . c
		k . c

$mk+1 \le n$

n variable
k constant

n > k
n < k
n, k

```
for( i=1; i<=n; i += k){
    for( j = 1; j<=k; j++){
        = } c
    }
}
```

$(m+1)\,k \cdot c \rightarrow \left(\frac{n}{k}+1\right) k \cdot c$

~

$mk+1 \leq n$	$\frac{nk}{k} \neq k \cdot c$
$mk \leq n-1$	$n \neq k$
$m \leq \frac{n-1}{k}$	n
$m = \frac{n}{k}$	$O(n) \Leftarrow$

i	1	k . c
i	1 k+1	k . c
i	2 k+1	k . c
i	3 k+1	k . c
	$mk+1 \leq n$	k . c

```
s=0
i=1

while(s<=n){
    s += i;
    i++;
}
```

c

m · c

$\sqrt{2}\cdot\sqrt{n}\cdot c$

$O(m)$

$O(\sqrt{n})$

$f(m) \le c \cdot g(n)$

$O(\sqrt{n}) \leftarrow$ max $\sqrt{n}$

s	1	1+2	1+2+3	1+2+3+4	$1+2+\dots+m \le n$
i	2	3	4	5	
	1	2	3	4	m

highest power

$$\frac{m(m+1)}{2} \le n$$

$$m(m+1) \le 2n$$

$$m^2 \le 2n$$

$$m \le \sqrt{2n}$$

$$m \le \sqrt{2}\cdot\sqrt{n}$$

```
for( i=1; i<= 1000; i++){
    =} c
}
```

i 1 c

i 2 c

⋮

i 1000 c

→ 1000 · c

$O(1)$

constant time

```
for(i=1;i<=n;i++){

    for( j = 1; j<= i; j++){
        = c
    }
}
```

i = 1 j 1

i = 2 j 2

i = 3 j 3

i = 4 j 4

⋮

i = n j n

$f(n) = 1+2+3+4 \cdots n-1+n$

$g(n) = n+n+n+n \cdots n+n$

$g(n) = 1 \cdot n(n)$

$= 1 \cdot n^2$

$O(n^2)$

```
for ( i=1; i<=n; i = i*2){
      = constant work
}
```

$2^m \leq n$

$m = \boxed{\log_2 n}$

$(m+1)c \quad \log_2 n \, c + c$

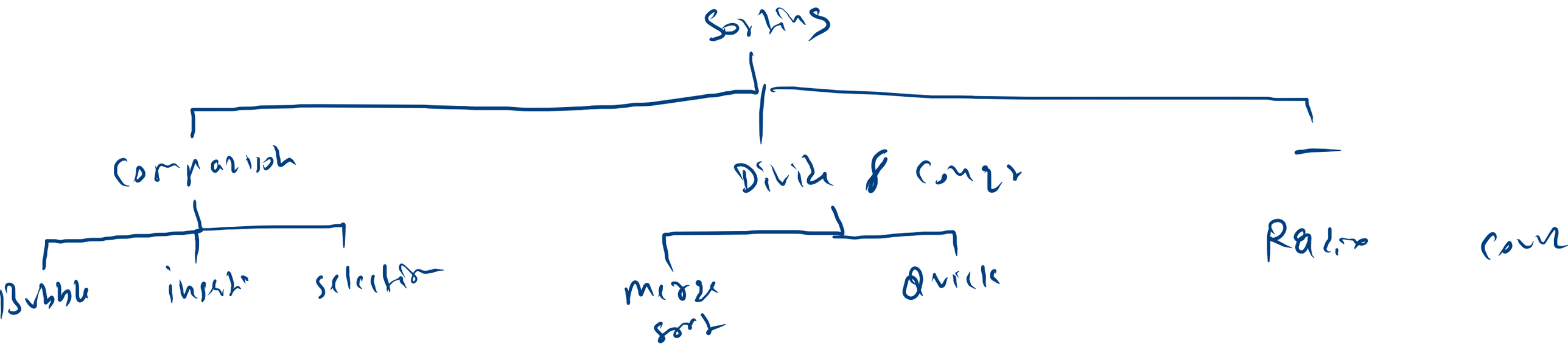
$O(\log_2 n)$ ∈

i	1	2^0
i	2	2^1
i	4	2^2
i	8	2^3
⋮	⋮	⋮
-	$\leq n$	2^m

const

O( )

m+1 times


Sorting
Comparison
Divide & conq
Bubble
insert
selection
merge sort
Quick
Radix
coun

n

1 iter

2 iter

3 iter

4 iter

n-1

n-1

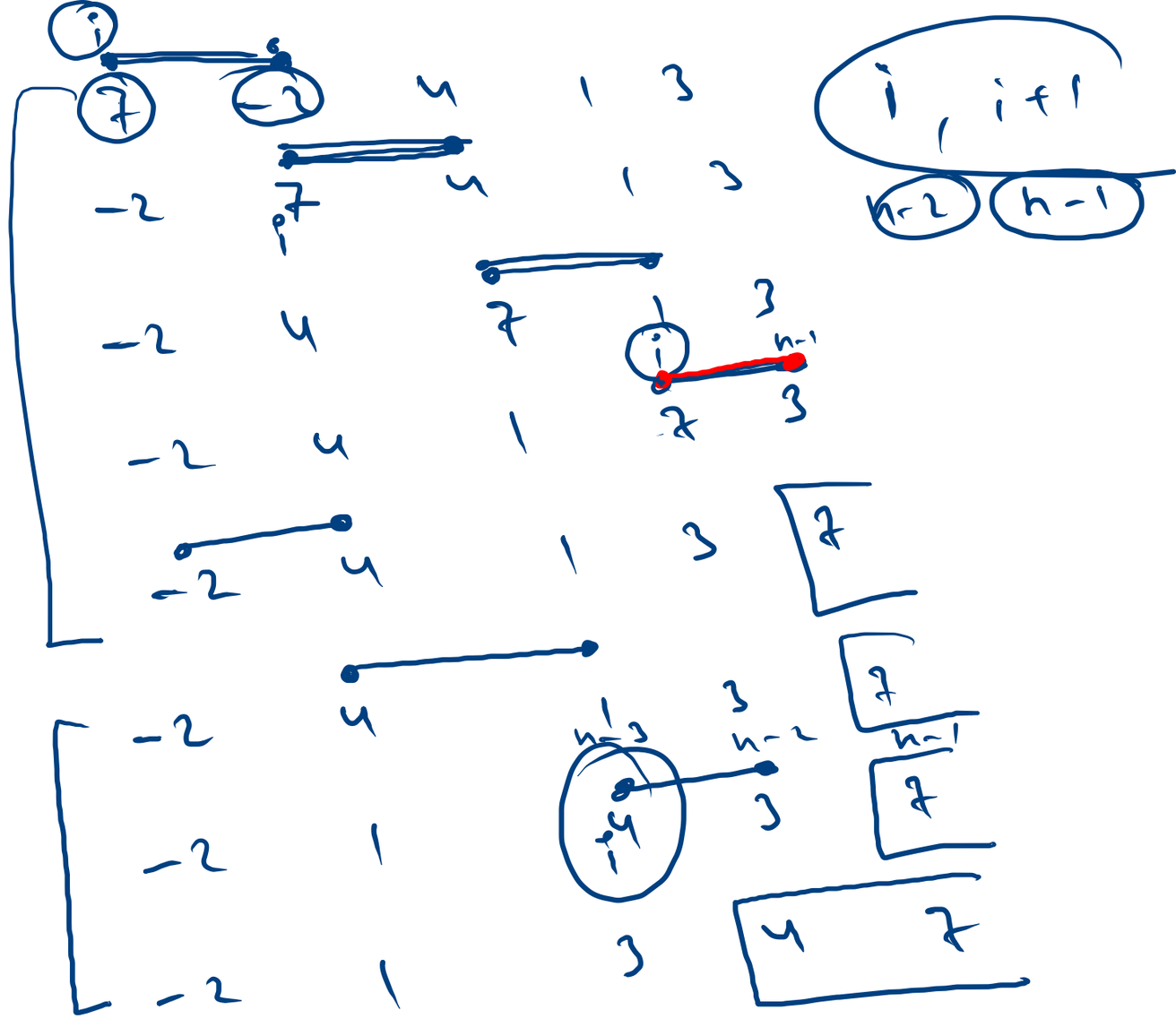
n

1 iter 0 .... n-2

2 iter 0 .... n-3

3 iter 0 .... n-4

iter 0 .... n-(iter+1) =



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

iter 1 → ⓪ … n-2 → [n-1] <

iter 2 → 0 … n-3 → [n-2] .<

0 … n-4 → [n-3] .<

3 → n-4

⋮ ⋮

n-1 → 0 → 1 .<

$\overline{(n-1) + (n-2) + (n-3) \ldots n-(n-1)}$

$n + n + n \ldots (n-1)\text{ times} - (1 + 2 + 3 \ldots + (n-1))$

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

$\boxed{O(n^2)}$ ←

$n-(1+1)$ $\quad$ $n-(n-1+1)$

$n-2$ $\quad$ $n-n=0$