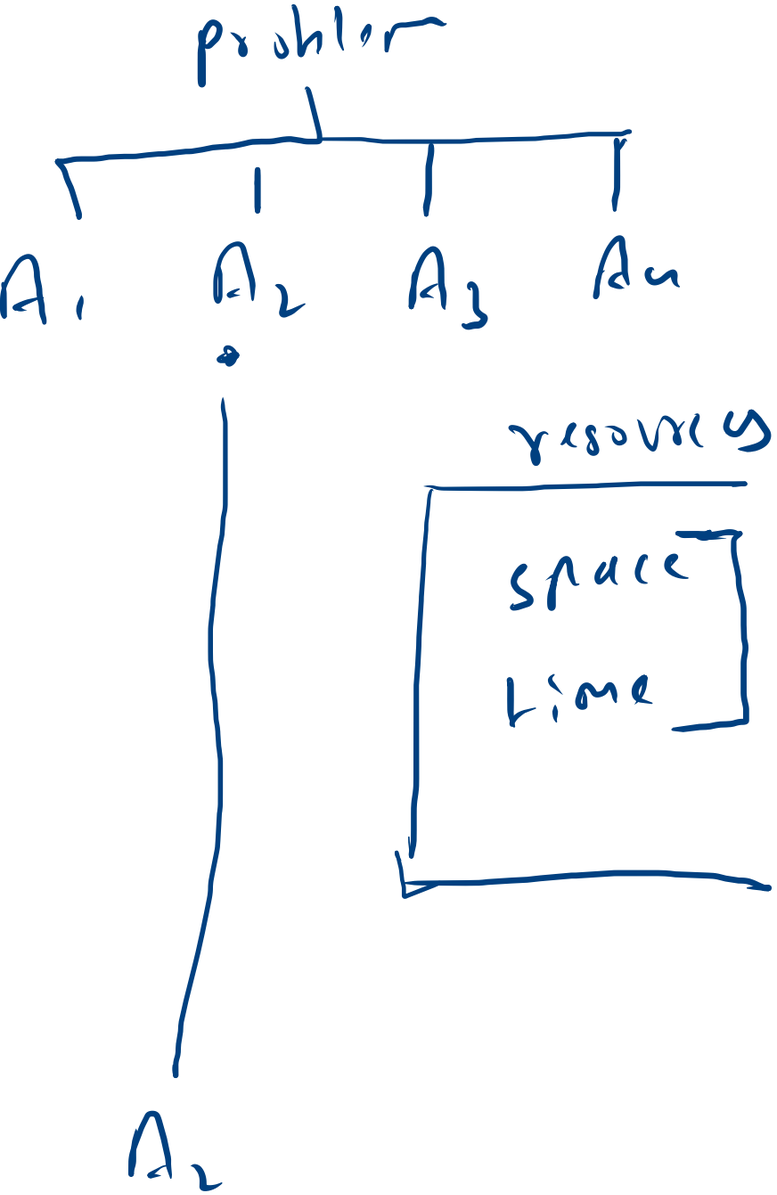
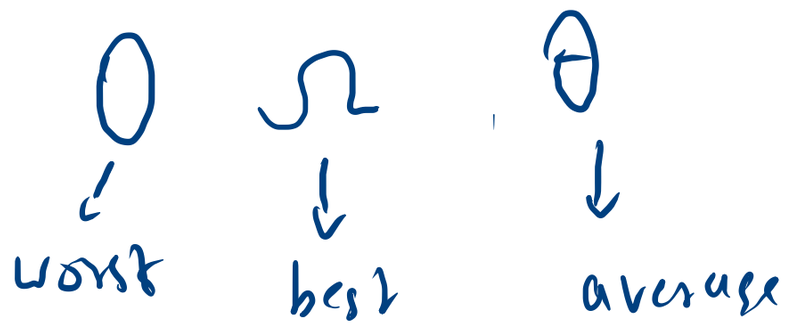
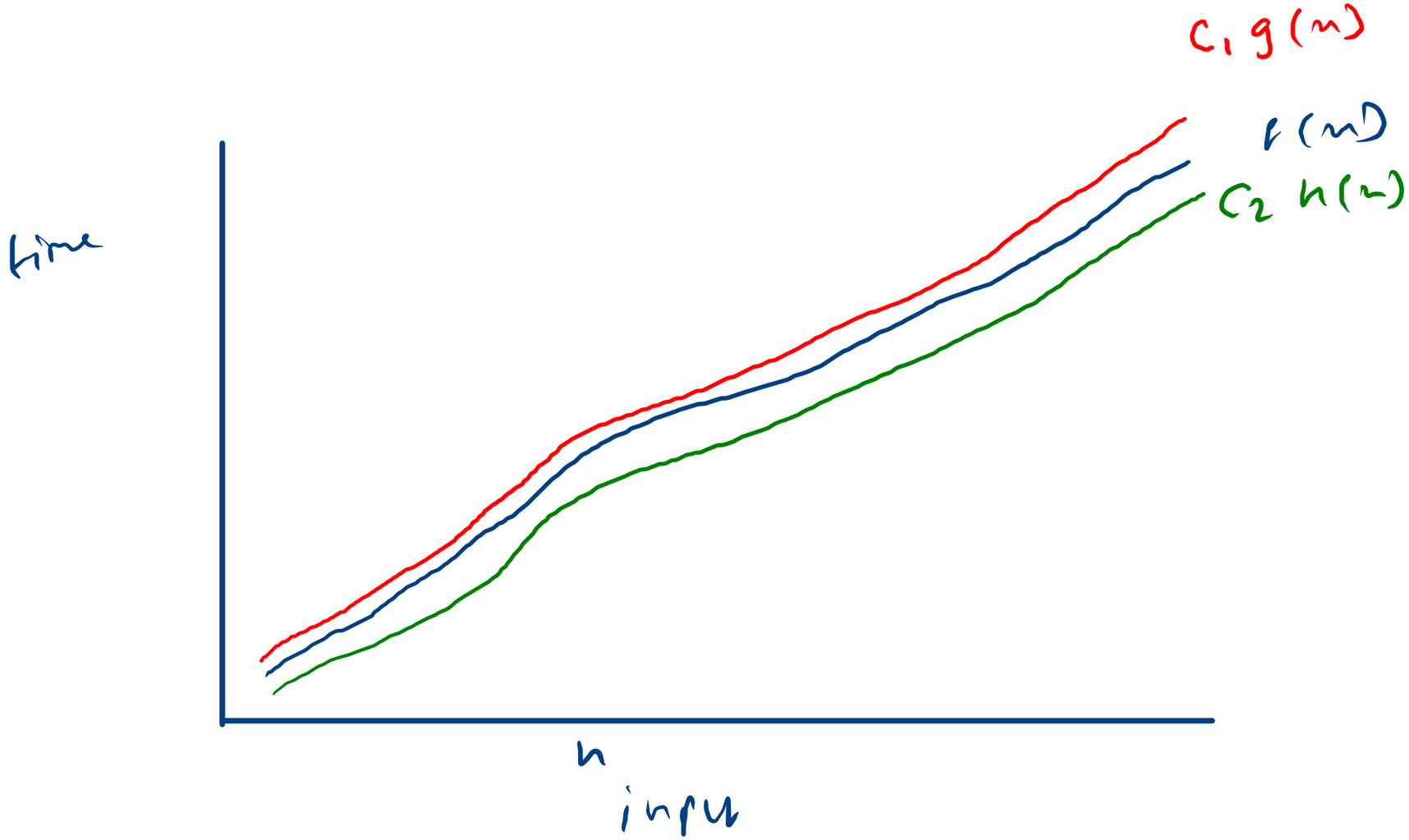

problem
A1
A2
A3
An
resources
space
time
A2


worst
best
average



$c_1 g(n)$

$f(n)$

$c_2 h(n)$

$f(n) \leq c_1 g(n)$

$f(n) = O(g(n))$

$f(n) \geq c_2 h(n)$

$f(n) = \Omega(h(n))$

Linear search

find 2

1

$\Omega(1)$

find 12

n

$O(n)$

$\theta\left(\frac{n}{2}\right)$

n = 10
n = 5
n = 1

$f(n) = 100$

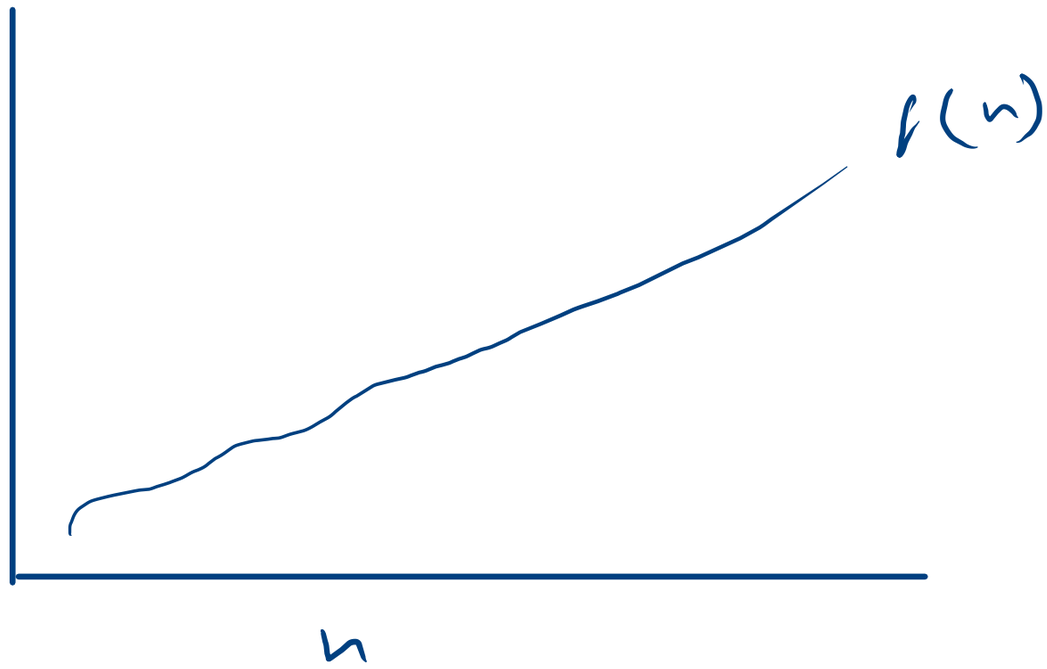
$c_1 g(n) = 100 \times 1$

$c_1 = 1$

$O(g(n))$

$O(1)$

100



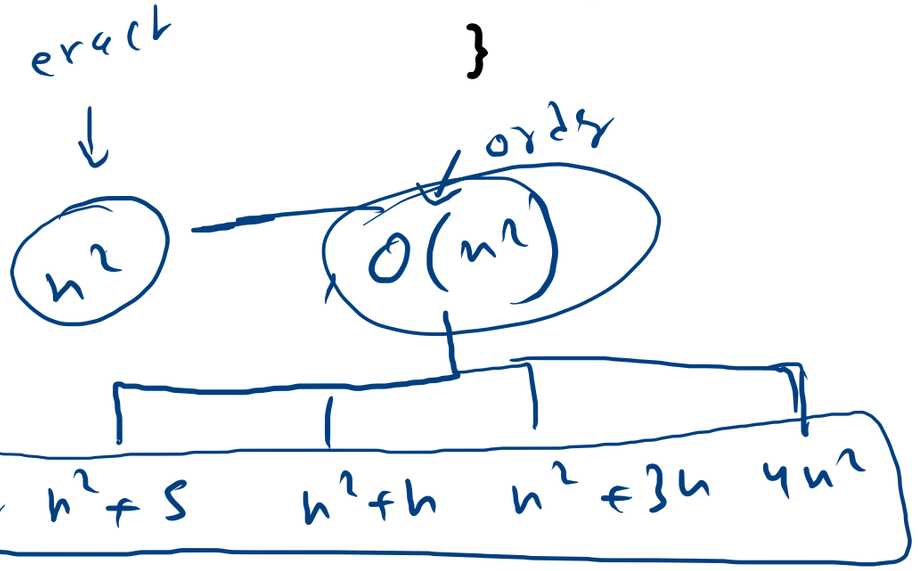

```
for( i=0; i<10; i++)
  {


}
```

3

O(1) ☆ ← const

```
for( i=0; i < n; i++){
    for( j=0;j<n;j++){
        }  c
    }
}
```



i	j	work
0	0 ... n-1	n x c
1	0 ... n-1	n x c
2	0 ... n-1	n x c
	⋮	
n-1	0 ... n-1	n x c

n times

$n \times c + n \times c + n \times c + \cdots$

$f(n) = n(n \times c)$

$n^2 \times c$

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

$\Rightarrow n^2 \times c$

$\Rightarrow O(n^2)$

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

$g(n)$

$\frac{c}{2}\left[n^2 + n\right]$

$\frac{c}{2}\left[n^2 + n^2\right] = \frac{c}{2}\left(2n^2\right)$

cn^2

$O(n^2)$

i	j	work
1	1	1 C
2	1..2	2 C
3	1..3	3 C
	⋮	
n	1....n	nC

$1C + 2C + 3C + 4C \cdots\cdots nC$

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

$f(n) = C\left[\frac{n(n+1)}{2}\right] = C\left[\frac{n^2}{2} + \frac{n}{2}\right]$

```
for( i =0; i<n; i = i*2){
}
```

i = i*2 ≡ i *= 2

i
0 → 0 → 0 → 0 → 0

while

```
{
{
{
{
```

∞

O(1) 7°/o

O(log n) 29<1.

O(n) 52 -1

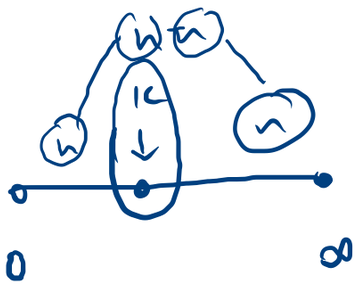
O(n²) 7-1.

≡ infinite [2] 5<1.

n variable
+k constant [0 — ∞]

k=n

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



O(1) 3

log(n) 6

n	55 ∠1
n^2	26
∞	10 ∠1.

$$\left[\left(\frac{n-1}{k}\right)+1\right] c$$

$$(n-1)\frac{c}{k} + \not{c}$$

n-1

n ⟸⟸ ← (n k)

O(n) O(n/k)

i

m	i	work
0	0 k+1	c
1	1 k+1	c
2	2 k+1	c
3	3 k+1	c
m	mk+1 ≤ n	c

$mk \leq n-1$

$m \leq \frac{n-1}{k}$

$m = \frac{n-1}{k}$

⇒ m+1

$f(m) = (m+1)c$

$f(m) = \left(\frac{n-1}{k}+1\right)c$

n variable
k constant

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

n^2

O(1) — O %

log(n) = 3 %

n = 50 %

n^2 = 22 %

nk = 25 %

i	j work
1 k	k c
→ 2 k	k c
→ 3 k	k c
⋮	
→ m k ≤ n	k c

$m \leq \frac{n}{k}$

$m = \frac{n}{k}$

m skoků

m k c

$\frac{n}{k} * c$

nc ← f(n)

f(n) = O(n)

```
i=0
s=0
while(s<=n){
      i++;
      s = s+i;
}
```

C

$n \rightarrow 5$ 2-1.

$n^2 \rightarrow 6$

$2n \rightarrow 3$

$n/2 \rightarrow 10$

1 →	9
$O(1)$	
$O(\sqrt{n})$	

i	1	2	3	4	m
s	1	1+2	1+2+3	1+2+3+4	$1+2\ldots m \leq n$

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

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

$$m(m+1) = 2n$$

$$\boxed{m^2 + m = 2n}$$

$f(m) = m$

$$m^2 + m = 2n$$

$$2m^2 = 2n$$

$$m^2 = n$$

$$m = \sqrt{n}$$

$O(\sqrt{n})$

$n^0 \quad n^1 \quad n^2$

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

$\sqrt{n}$

$n^{1/2}$

$n^{1/3}$

n

$m = \log_2 n$

$f(n) = (m+1)c$

$= c\log_2 n + c$

$g(n) = k\log_2 n$

$O(\log_2(n))$

$\log_2(n)$	(1s) n	
2^0	$1 \leq$	1s
2^1	$2 \leq$	1s
2^2	$4 \leq$	1s
2^3	$8 \leq$	1s
2^4	$16 \not\leq$	1s

$m+1$ → i

2^0	1	c
2^1	2	c
2^2	4	c
2^3	8	c
⋮	⋮	

$2^m \hookrightarrow n \leq n$

$2^m \leq n$

$2^m = n$

c

$2^m = n$

$\log_2 2^m = \log_2 n$

$m \log_2 2 = \log_2 n$

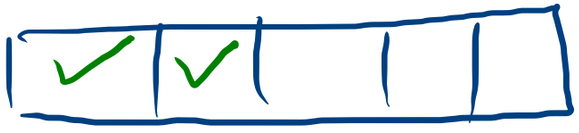
Order ∈

7 -2 4 1 3 ⟶ -2 1 3 4 7

1 iter

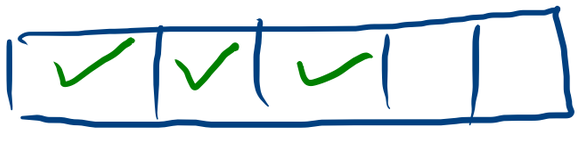
-2

2 iter



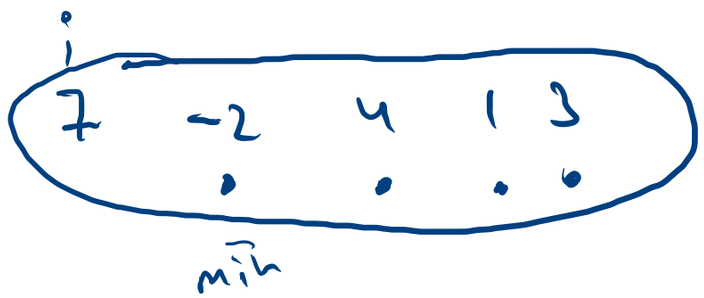
-2 1

3 iter



0 1

7 -2 4 1 3



4 ith → -2 1 3 4 7

1 ith → -2 7 4 1 3

min

if (issmaller(arr, i, j)) {

swap(arr, i, j);

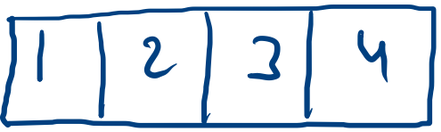
}

2 ith → -2 1 4 7 3

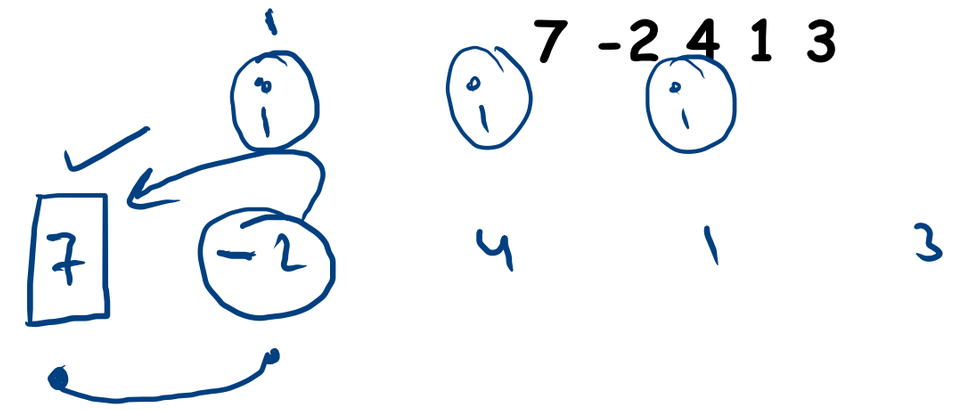
min

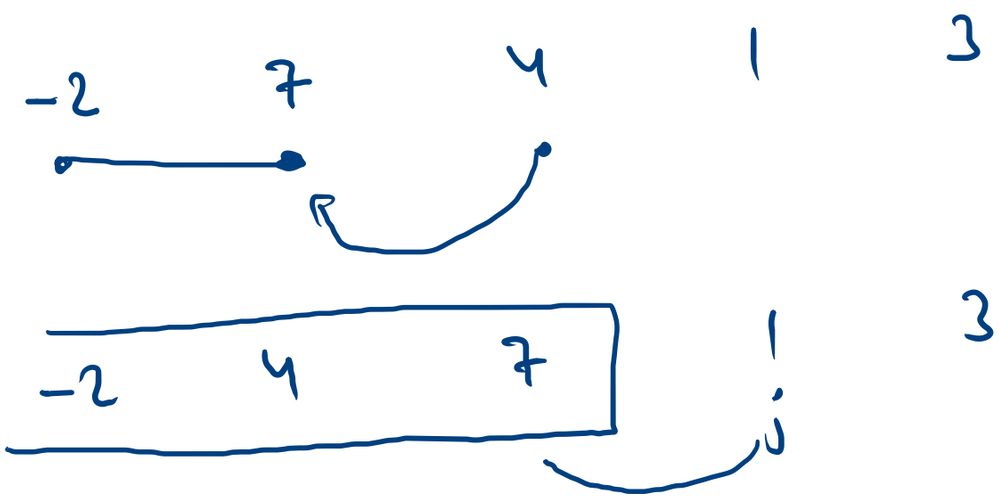
3 ith → -2 1 3 7 4

n → n-1

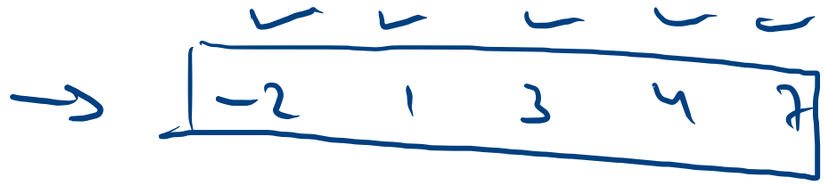

1
2
3
4

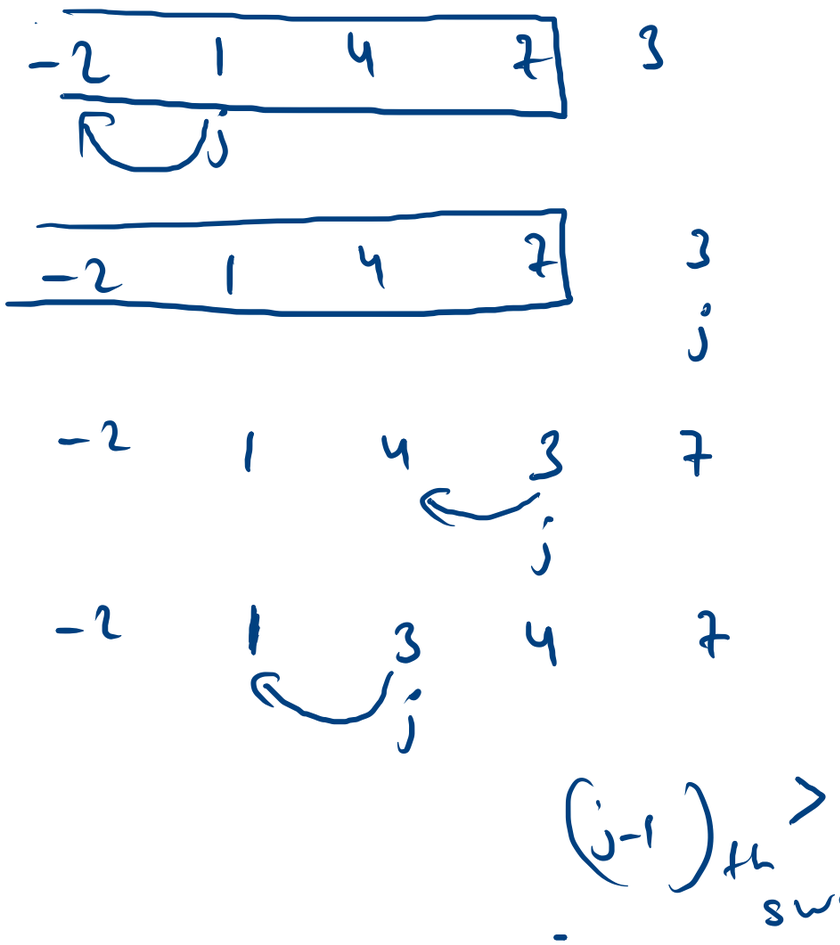
7 -2 4 1 3


i
i
i
7
-2
4
1
3


-2
7
4
1
3
-2
4
7
j
3
-2
4
1
7
3


-2
4
1
7
3
j


-2
1
3
4
7


-2
1
4
7
3
j
-2
1
4
7
3
j
-2
1
4
3
7
j
-2
1
3
4
7
j
(j-1)th > jth
swap

1+2+3+4

4×5/2

```
for(int i=1;i<n;i++){
    int j = i;
    while(j-1 >= 0 && isGreater(arr, j-1, j)){
        swap(arr, j-1, j);
        j--;
    }
}
```

1 2 3 4 5 ...

worst case

2 3 4 5 ... n

i

1 → k

2 → k

3 → k

⋮ → k

n-1 → k

$O(n)$ nk

i

1 → k

2 → 2k

3 → 3k

⋮

→ (n-1)k

$\left[\frac{(n-1)\,n}{2}\right]k = O(n^2)$

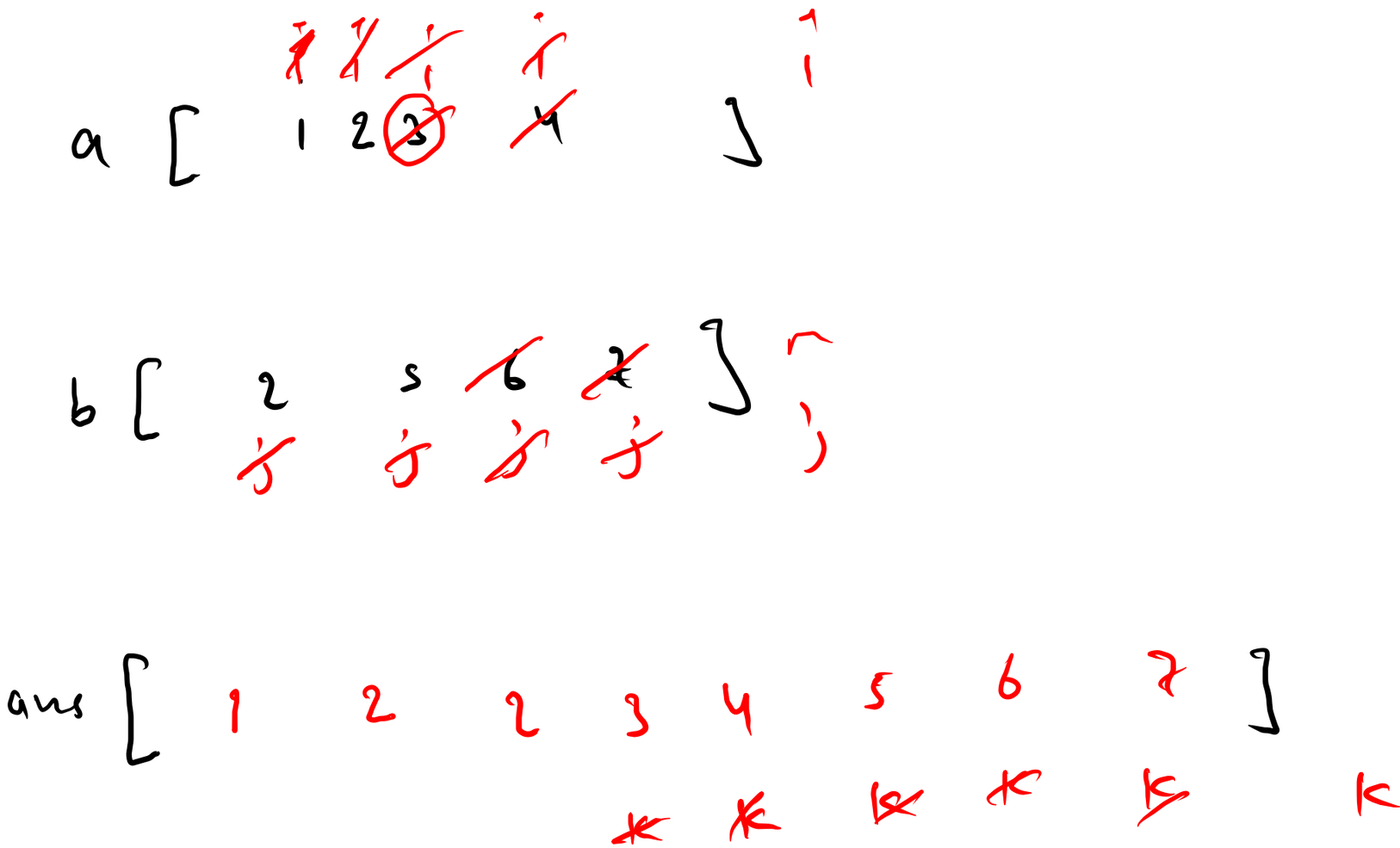
bubble → $O(n^2)$ $\Omega(n^2)$

selection → $O(n^2)$ $\Omega(n^2)$

insertion → $\Omega(n)$ $O(n^2)$

```
while(i<n && j<m){
    if(a[i] < b[j]){
        ans[k] = a[i];
        i++;
    }else{
        ans[k] = b[j];
        j++;
    }
    k++;
}

while(i<n){
    ans[k] = a[i];
    i++;
    k++;
}
while(j<m){
    ans[k] = b[j];
    j++;
    k++;
}
```



Solution → S2 → optimize → optimize → linear

a = [1 3 3 7 8 9] n

b = [2 4 5 11 12 13] m

i

j

[1 2 3 3 3 4 5 7 8 9 11 12 13]

a = [4 5 6]

b = [1 2 3]

[

O(n) O((n+m)^1) O(n+m)

Sol1 a -> [1 3 3 7 8 9] n = 6

i

Sol1 b -> [2 2 4 5 11 12 13] m

j

K

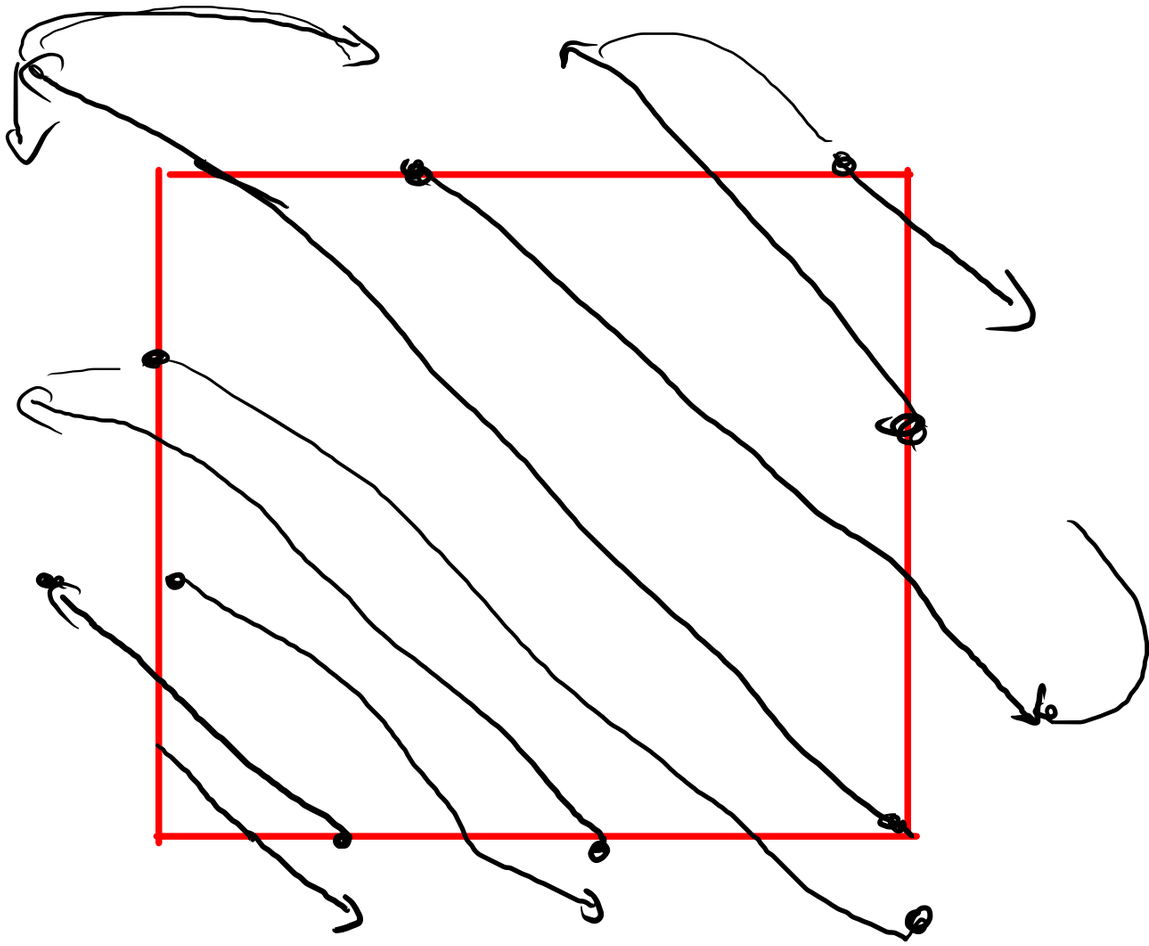
ans [1 2 3 3 7 4 5 7 8 9 11 12 13]

Sol2

$O(n+m + (n+m)^2) \rightarrow O((n+m)^2) >> O(n+m)$

Lah

- $O(1)$
- $O(\log(n))$
- $O(\sqrt{n})$
- $O(n)$
- $O(n\log(n))$
- $O(n^2)$
- $O(2^n)$



```
loop {
    ↘
    ↖
}
```

1 → 9

2 → 12

3 → 0

4 → 0

5 → 3

(

,

9 → 0

1 >

same class

1 class

1 class

Iteration

Recursion

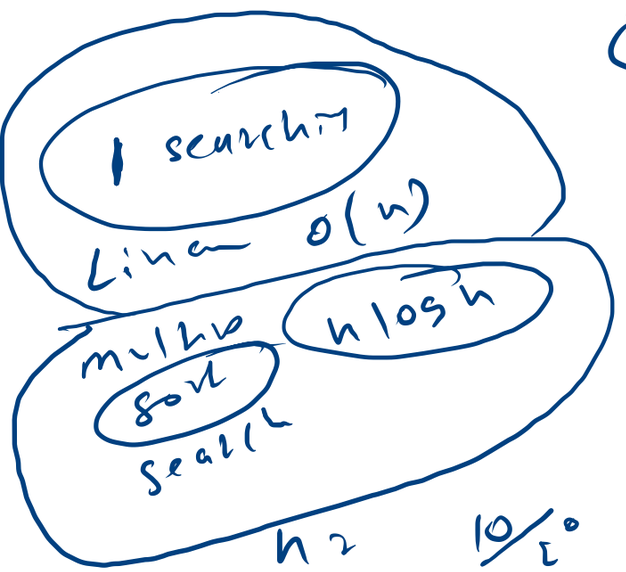
100

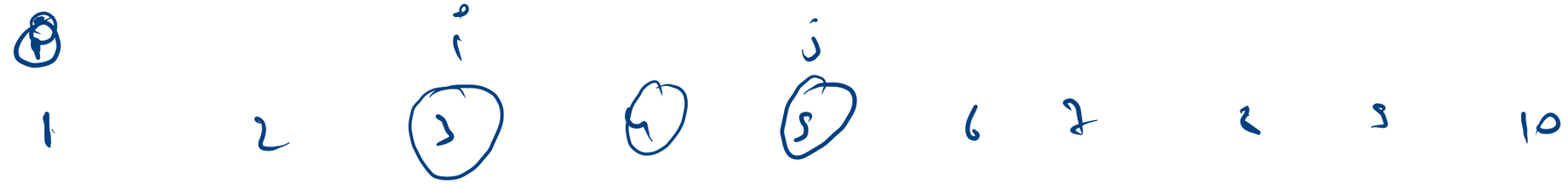
for(int i=0; i<=100;i++)

=

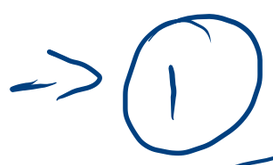
=

}





			work
n_2	$10/i^0$	$\frac{10}{2^0}$	K
n	$\frac{10}{2}$	2^1	K
n	$\frac{10}{n}$	$\frac{10}{2^2}$	K
	$\frac{10}{8}$	$\frac{10}{2^3}$	
n	1	$\frac{10}{2^m}$	



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

$$2^m = n$$

$$m = \log_2(n)$$

$$O(n) = \log_2(n) + 1$$

$$\boxed{O(\log_2(n))}$$