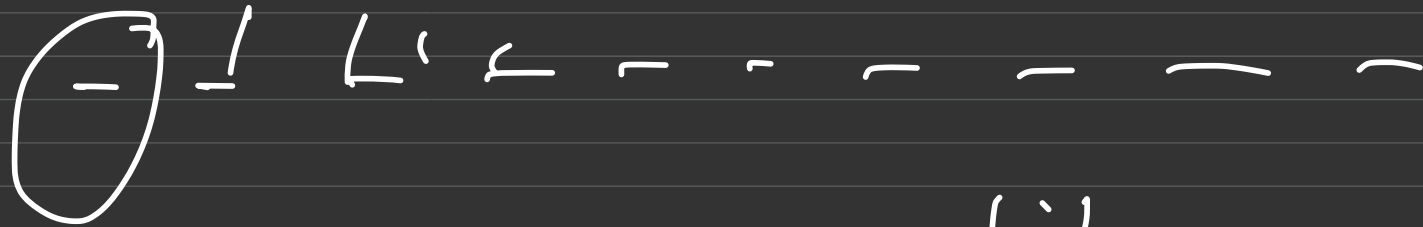




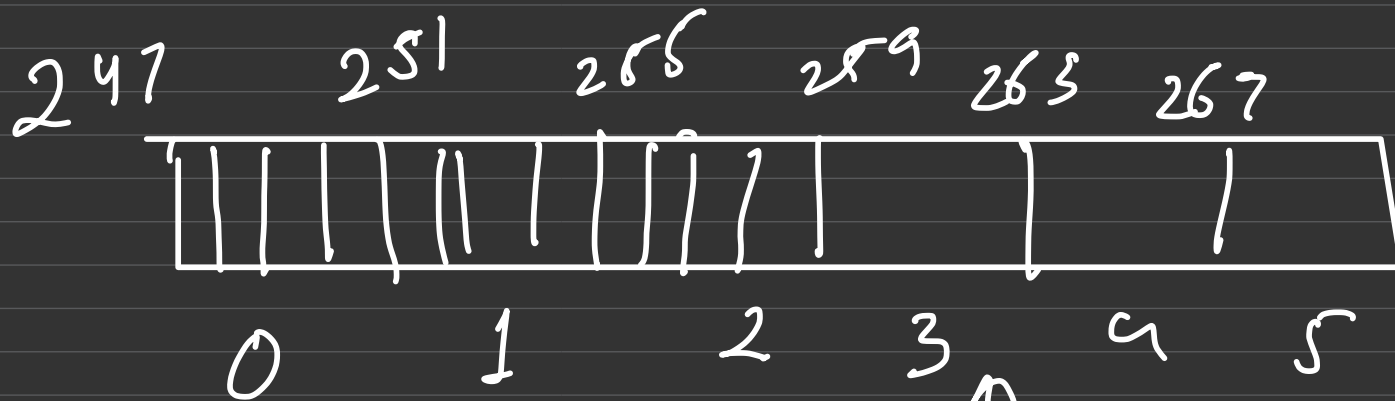
int a[1000] int a;

$$2^{32} - 1$$



bits

1 byte = 8 bits

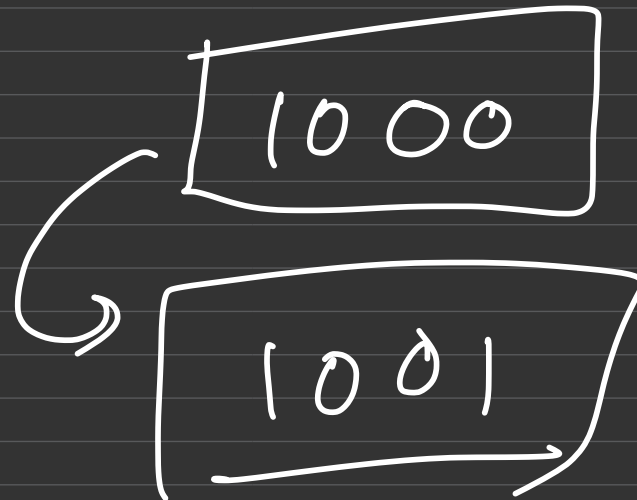


arr[3]



$$247 + 4 \times \underline{3}$$

0 ()



$\theta(n)$

$\theta(n)$

1	2	3	1	1
---	---	---	---	---

Elc

Size

6
Capacity

0

0

0

1

5 →

1

1

6 →

2

2

8 →

3

4

11 →

4

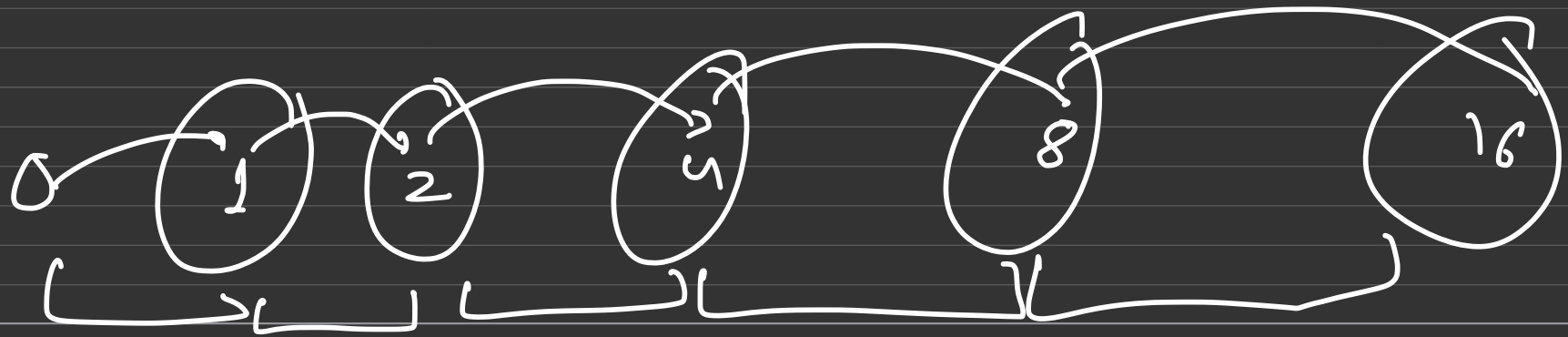
4

12 →

5

8

Elc	Size	Cap
18	6	8
20	7	8
25	8	8
40	9	16



η η

0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128

16⁷

256

\downarrow

~~512~~

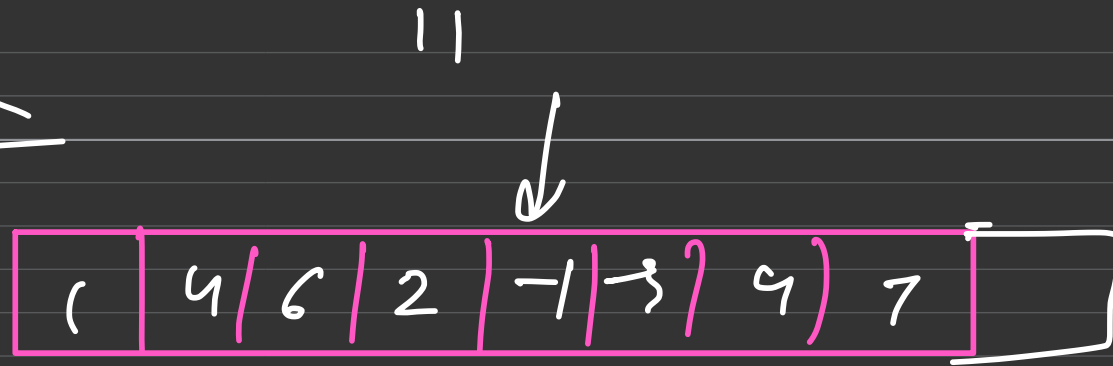
\downarrow

8192 \leftarrow 4096 \leftarrow 2048 \leftarrow 1024

16⁴

16-17

Search



Insert at the end

Insert at the start

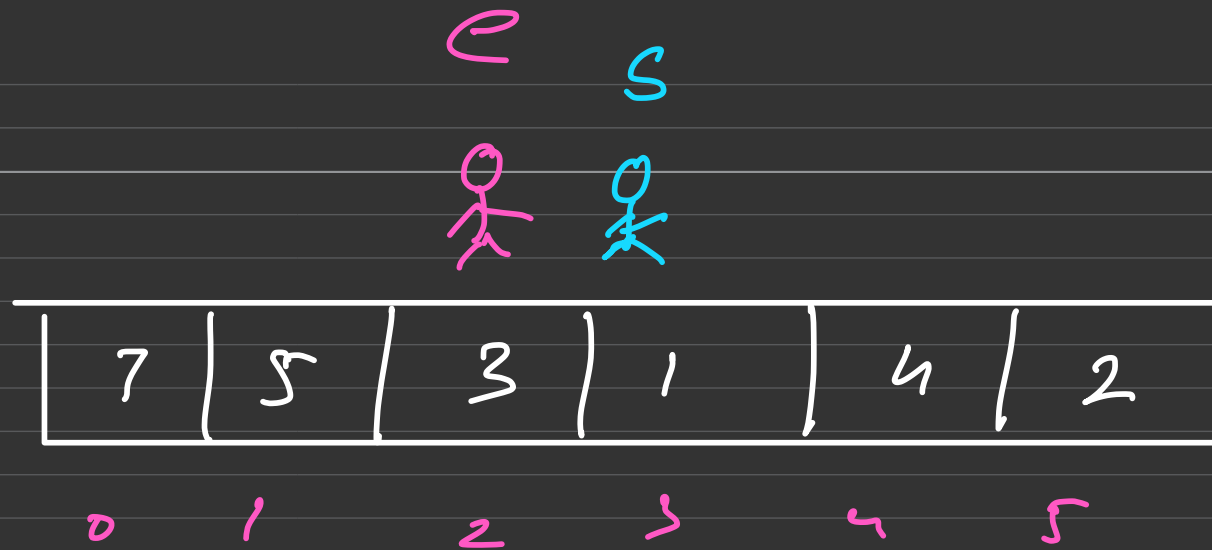
$$\frac{20}{10^7} \times 100$$

→ int j = 0;



1	3	12	0	0
0	1	2	3	4

```
for (int i = 0; i < n; i++)  
{  
    if (arr[i] != 0)  
    {  
        swap(arr[i], arr[j])  
        j++;  
    }  
}
```



while ($s < e$)

{ swap($arr[s], arr[e]$);

$s++$;

} $e--$;



1	2	6	3	0	7
---	---	---	---	---	---

max_val = 7



1 2 1 4 6

min_val = 1

sec_min_val = 2

```
min_val = max;  
sec_min_val = max;
```

```
for (int i=0; i < n; i++)
```

```
{
```

```
    if (ar[i] <= min_val)
```

```
    {  
        sec_min_val = min_val;  
        min_val = ar[i];  
    }
```

```
    else if (ar[i] < sec_min_val)
```

```
    {  
        sec_min_val = ar[i];  
    }
```

```
}
```

1 →

2	3	1	2	4	0	8
---	---	---	---	---	---	---

$O(n)$

2 →

2	1	2	3	0	4	8
---	---	---	---	---	---	---

1 2 3 4 5

$n-1$

int i = 1; i <= n-1; i++

{

for (int j = 0; j < n-i; j++)

if (arr[j] > arr[j+1])

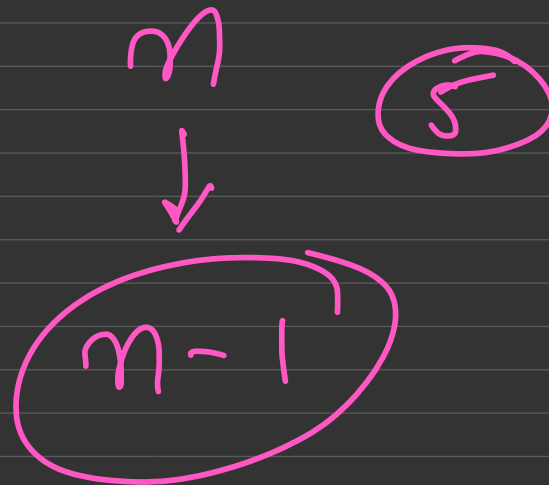
swap(

→

)

Worst

$$\underline{\underline{O(n^2)}}$$



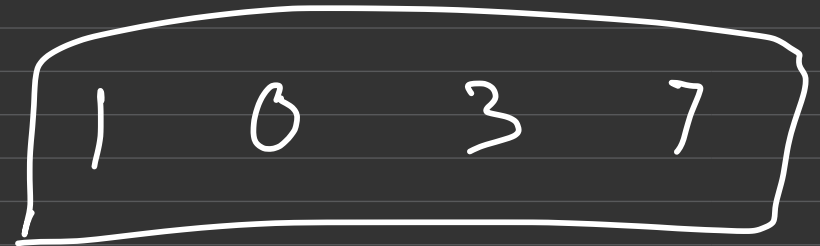
$$n^2$$

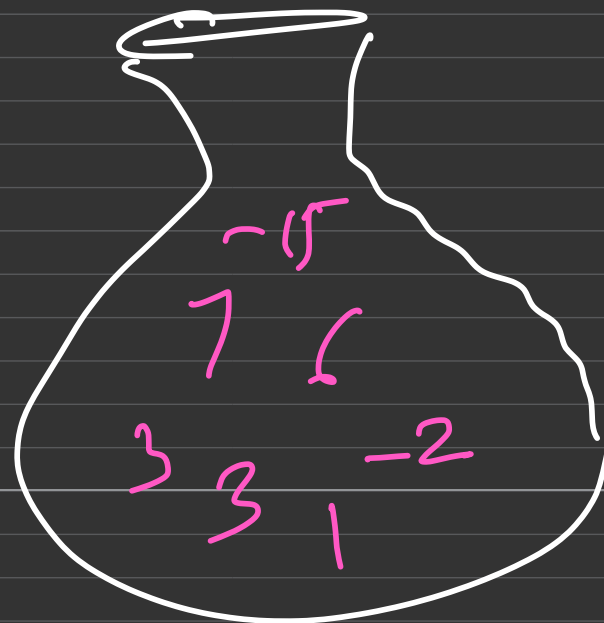
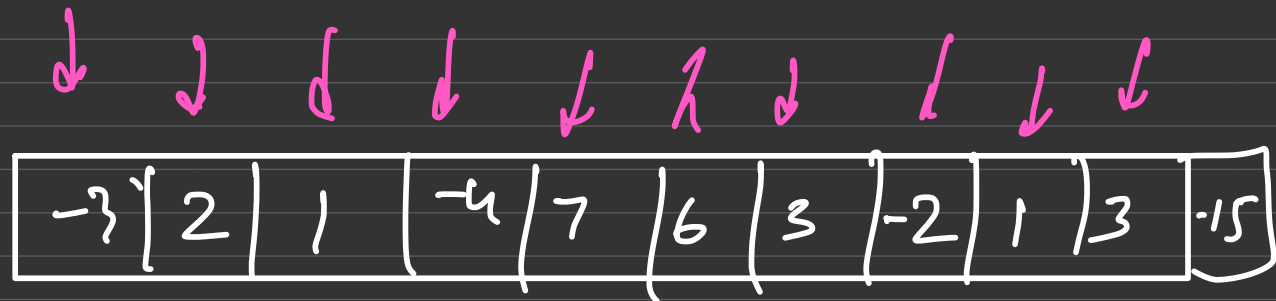
$$\underline{\underline{O(n)}}$$



Subarray

Subsequence





Sum

-3 14
 2 15
 3 18
 -1 3
 7
 13
 16