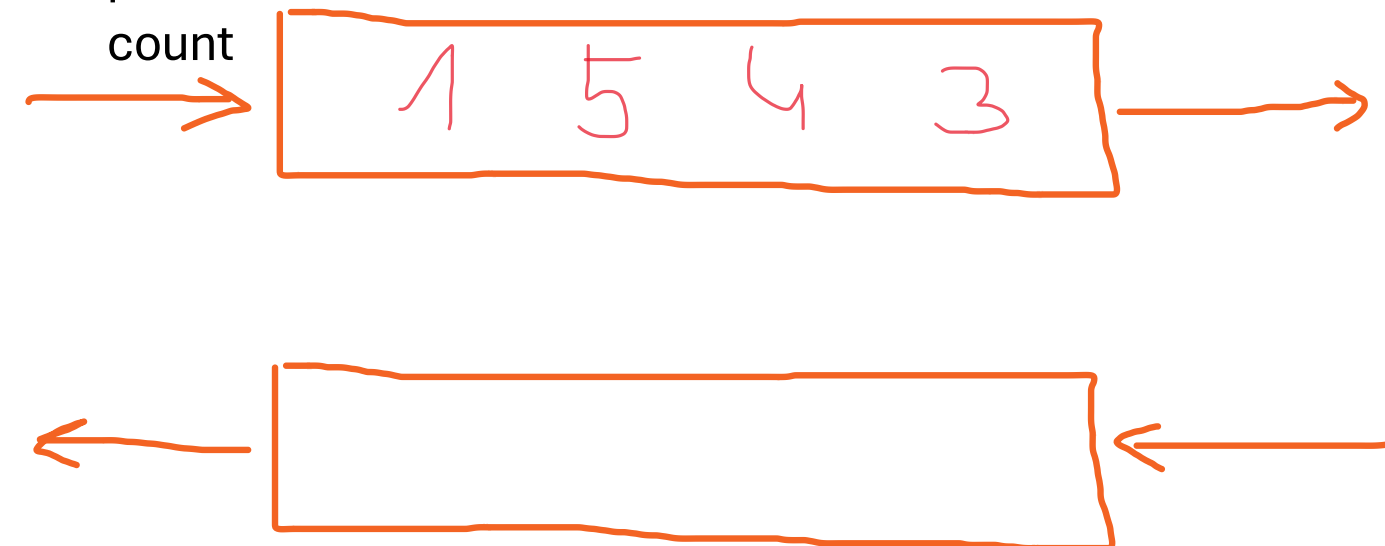


enqueue  
dequeue  
peek  
count



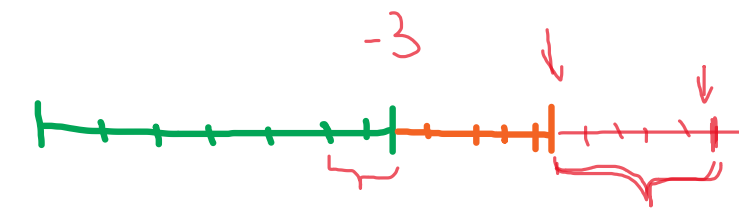
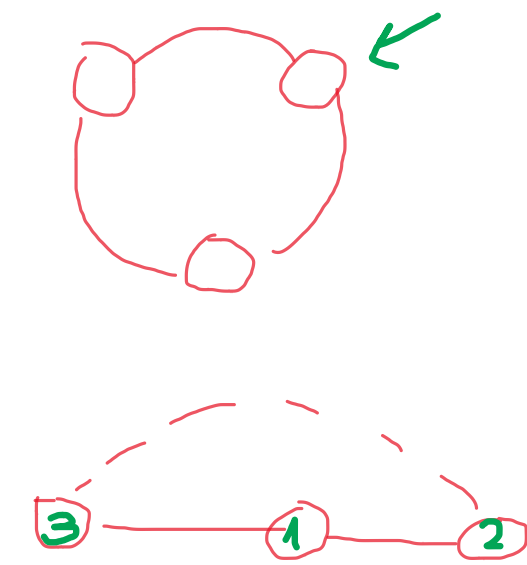
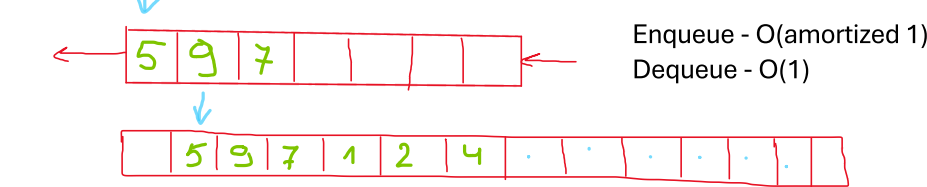
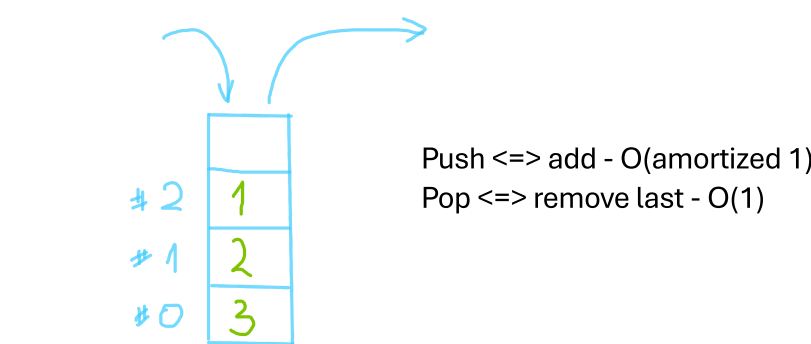
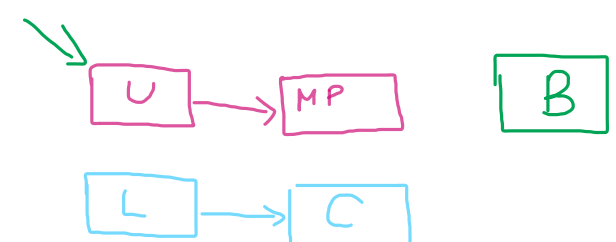
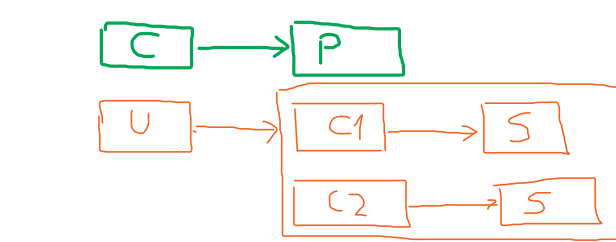
Search:

- Array/List =>  $O(n)$
- Set/Dictionary =>  $O(1)$
- SortedSet/SortedDictionary =>  $O(\log n)$

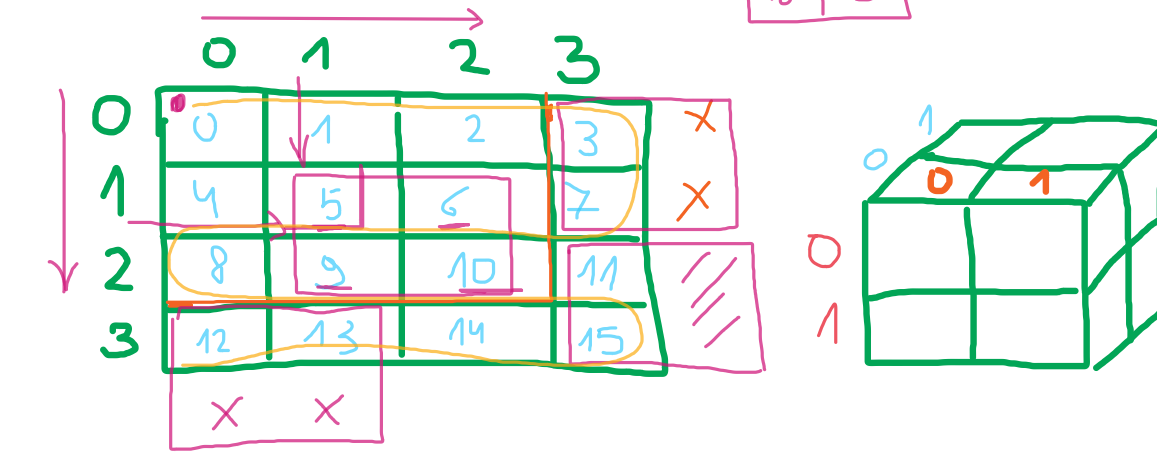
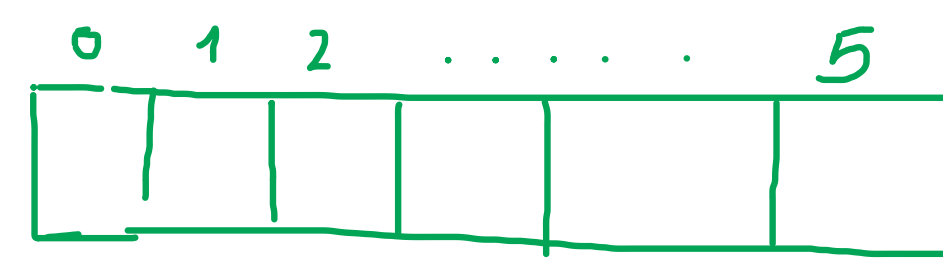
Add:

- List =>  $O(1)$
- Set/Dictionary =>  $O(1)$
- SortedSet/SortedDictionary =>  $O(\log n)$

Add -  $O(\text{amortized } 1)$   
Contains -  $O(n)$   
Swap -  $O(1)$   
RemoveAt -  $O(n)$   
Insert -  $O(n)$



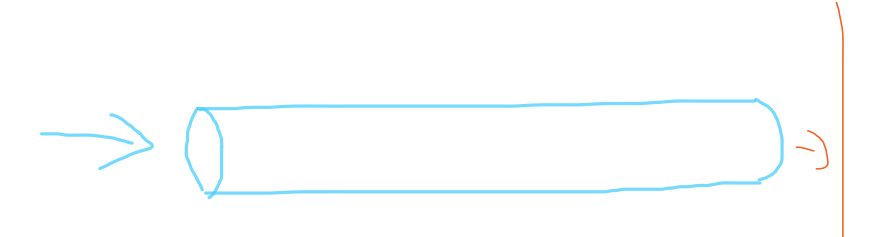
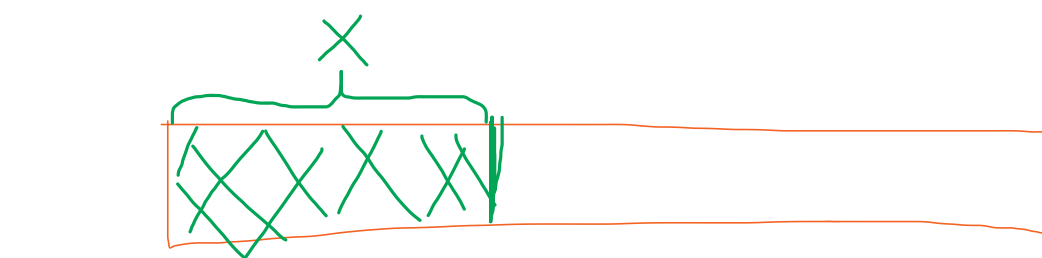
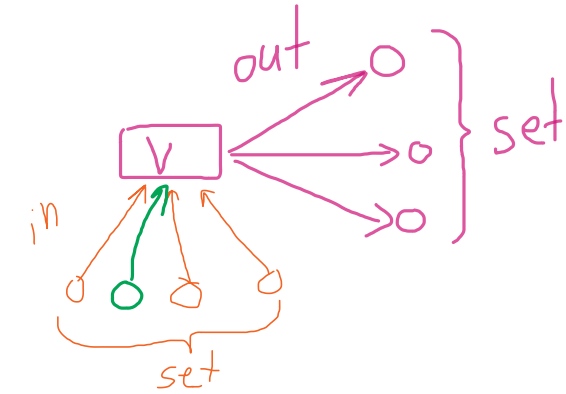
$$2 - 3 = -1$$



$$4, 4 \Rightarrow 16$$
$$16 \neq \dots$$

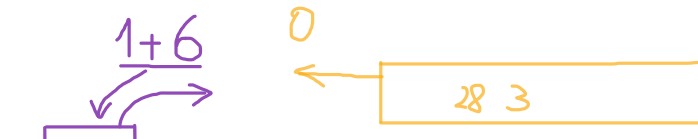
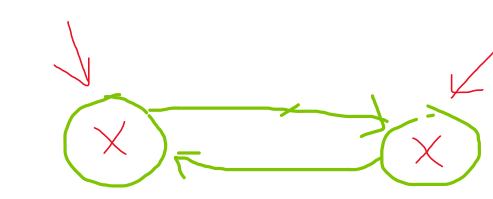
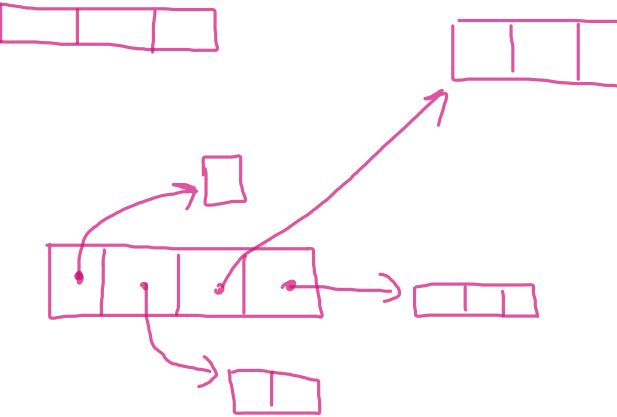
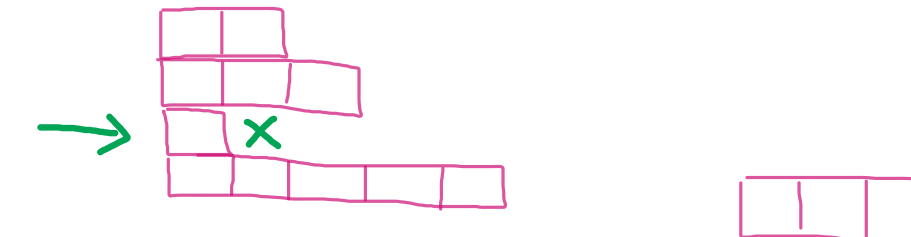


$$10 / 4 = 2$$
$$10 \% 4 = 2$$

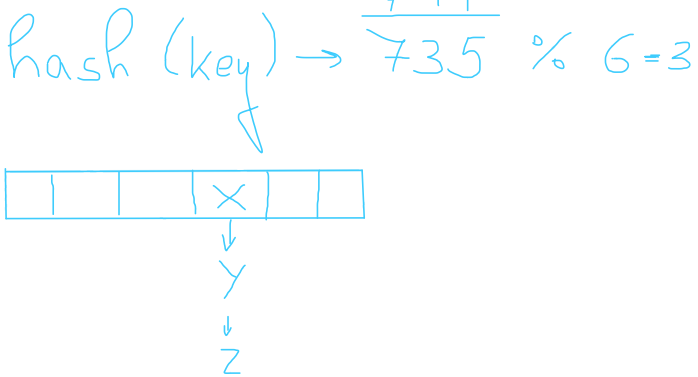
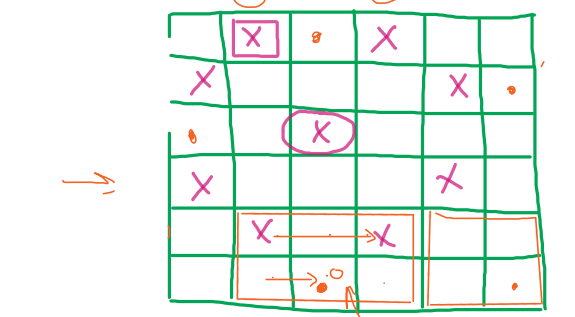
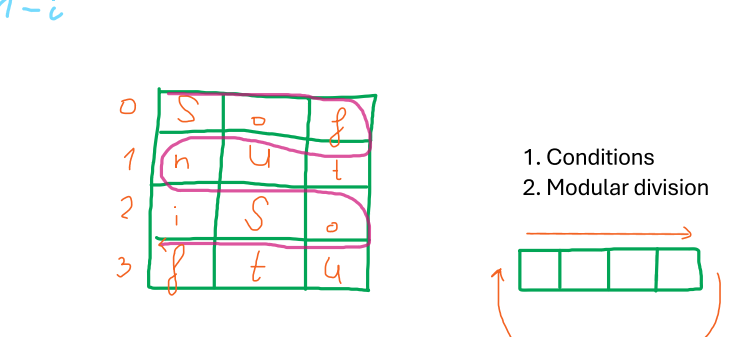
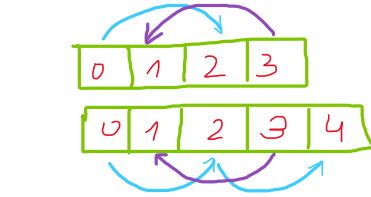
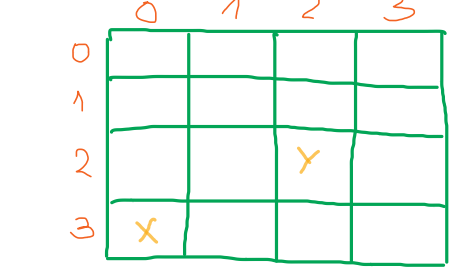
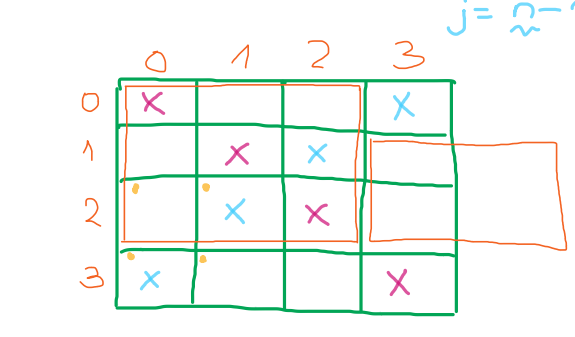
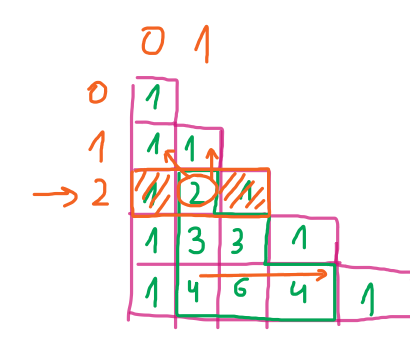


```
long[,] image2D = new long[640, 860];  
byte[,] image3D = new byte[640, 860, 3/4];  
  
// 60 * 30 = 1800  
long[,] video3D = new long[1800, 640, 860];  
byte[,,,] video4D = new byte[1800, 640, 860, 3/4];
```

[2][4]



$$32 / 7 = 4 \text{ (4)}$$
$$50 / 7 = 7 \text{ (1)}$$



1. Conditions  
2. Modular division

$n = 30$   
 $k = 900$

Brute force:  $O(2^{900})$   
Greedy:  $O(k \log(k)) \rightarrow O(k^2)$   
Union-Find | DSU:  $O(k)$

$$\text{hash}(\text{key}) \rightarrow 735 \% 6 = 3$$