

$i=j$

	0	1	2
0	11	3	4
1	4	5	6
2	10	8	12

primary diagonal
sum = 11 + 5 + 12 = 4

$j = n - (i + 1)$

	0	1	2
0	11	2	4
1	4	5	6
2	10	8	-12

secondary diagonal
sum = 4 + 5 + 10 = 19

3 4

A	B	B	0
E	B	B	B
1	3	B	0

1	5	5	1
2	1	4	4
3	7	14	4
4	8	12	4



0K0K0	L	1,2	L	2,1
K000K	└	-1,2	└	2,-1
00K00	┐	-1,-2	┐	-2,1
K000K	└	-1,-2	└	-2,-1
0K0K0	┐	1,-2	┐	-2,-1

** * c *	↑	-1,0
** * e *	→	0,1
** * c *	↓	1,0
s * c *	←	0,-1

FIFO

4 32 45

- HashSet & Dictionary
- hash table
 - No indexing
 - Order is not guaranteed
 - Uniqueness
 - Fast - O(1)

- SortedSet & SortedDictionary
- BST
 - No indexing
 - Order is guaranteed
 - Uniqueness
 - Relatively fast - O(log n)

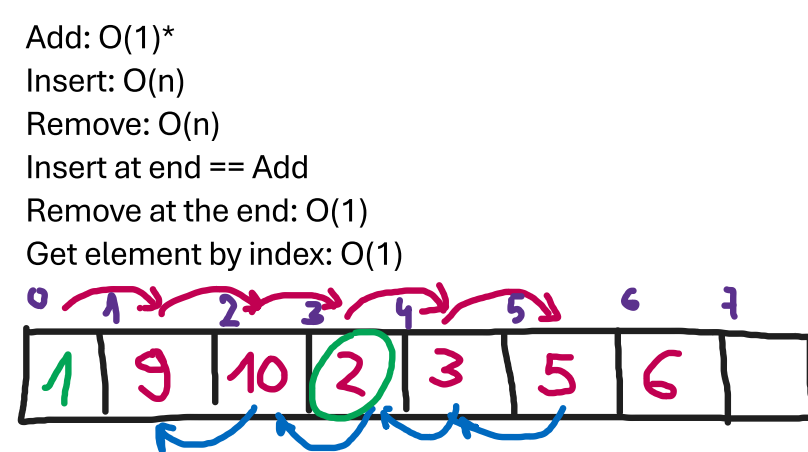
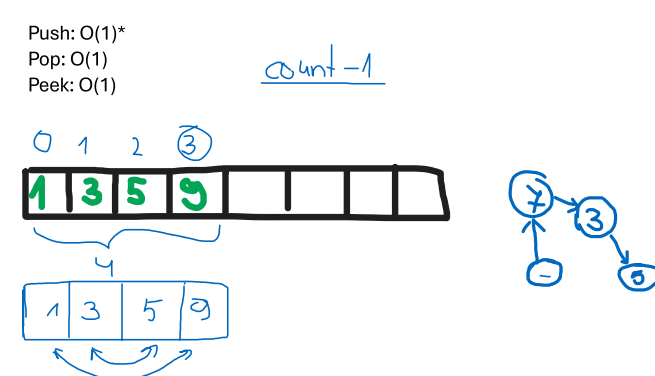
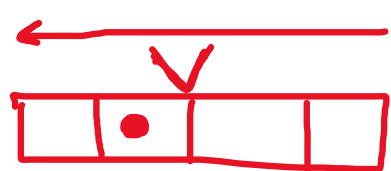


ACC (el, acc) → new-acc

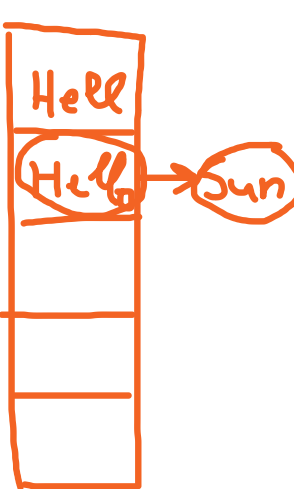
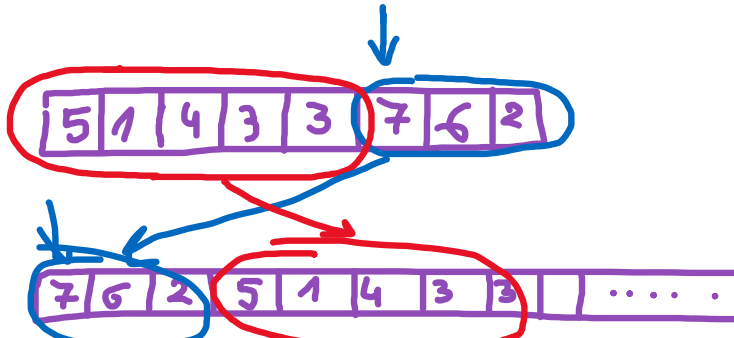
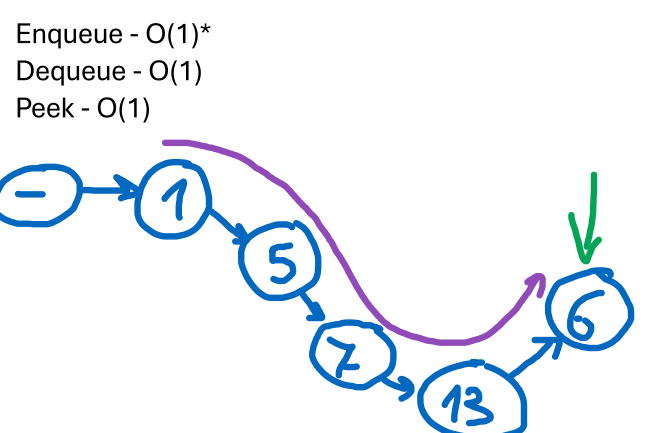
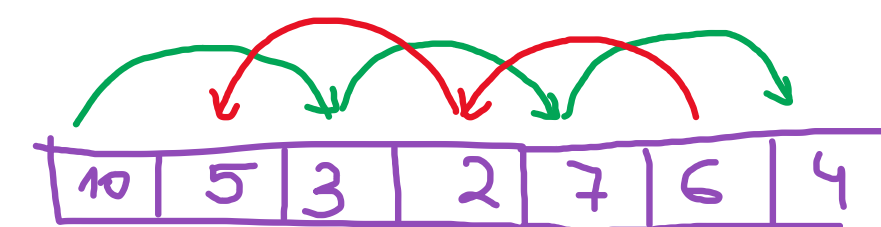
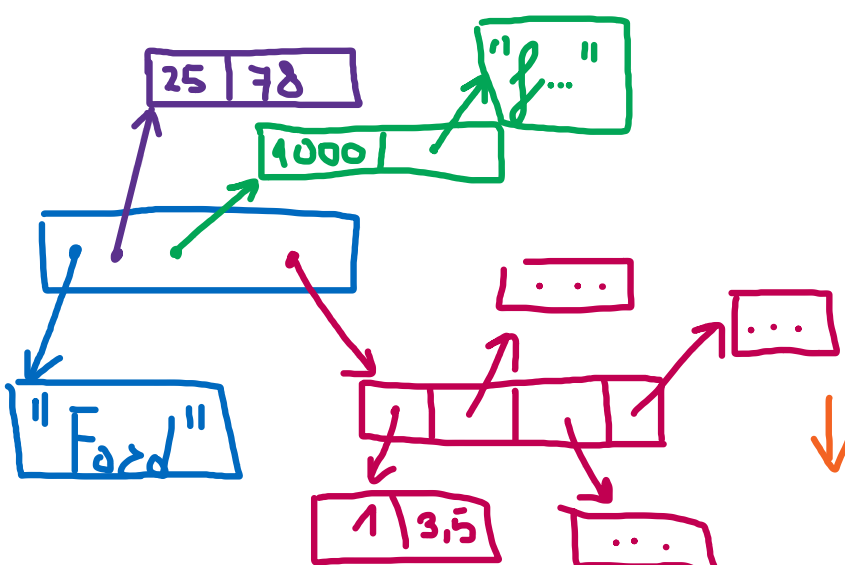
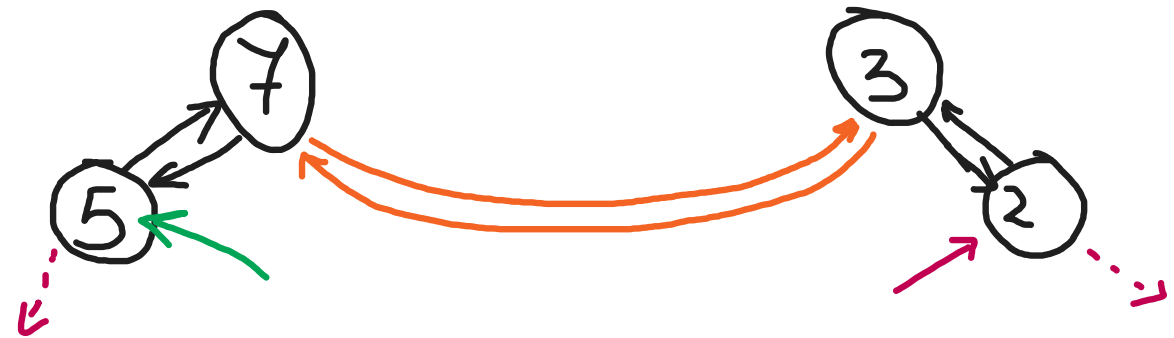
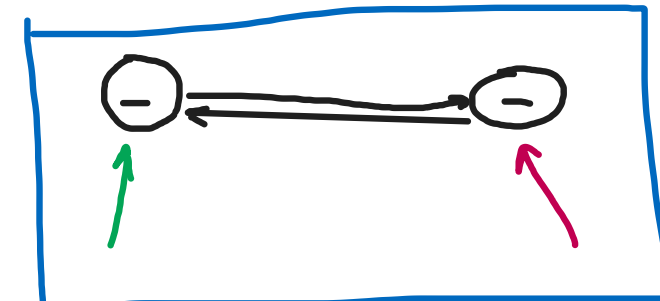
1 5 3 6



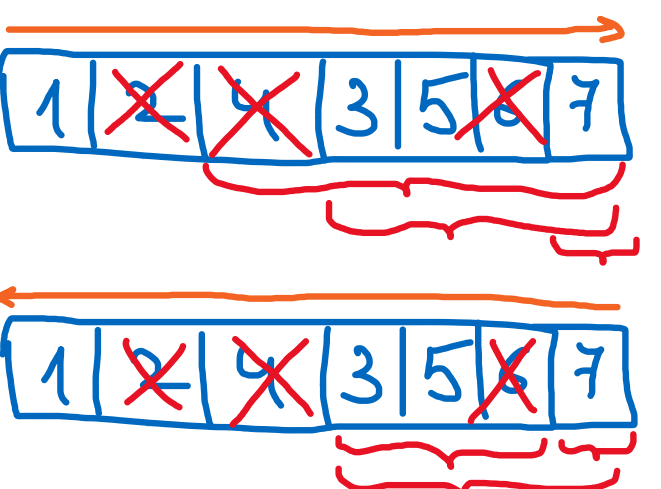
CreatePrinter()
↳ (p) → void



Add front/back: O(1)
Insert in the middle: O(n)
Remove front/back: O(1)
Remove in the middle: O(n)
Get element by index: O(n)



Hello → 1351176 % 5 = 1
Hell → 27200 % 5 = 0
Sun → 6 % 5 = 1



.ToString()
→ "....."
.Trim()
→ "....."