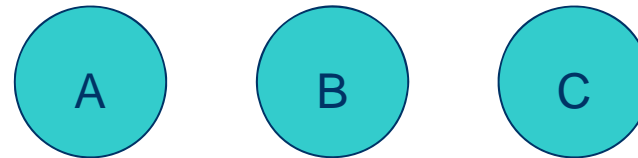


ADT Sequence & its implementation

- Operations

- Create / remove
- Add / remove element
- Membership (T|F)
- Cardinality (# elements)
- Is_empty: $S \rightarrow T|F$
- Is_pos_valid: $S \times p \rightarrow T|F$
- Find element at position p
 - Find: $S \times p \rightarrow e$
- Find position of an element
 - Find: $S \times e \rightarrow p$
- Find first / next / last element
- Find first / last position

- Abstract sequence

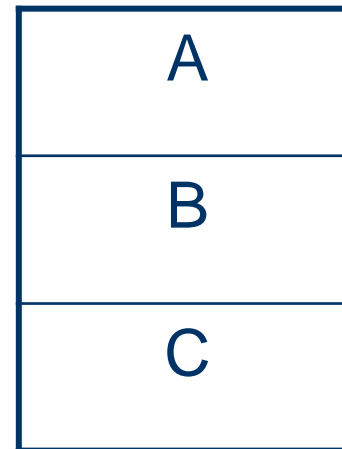
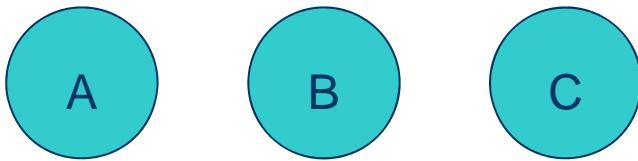


- Properties

- Successor
 - Except the last
- Predecessor
 - Except the first

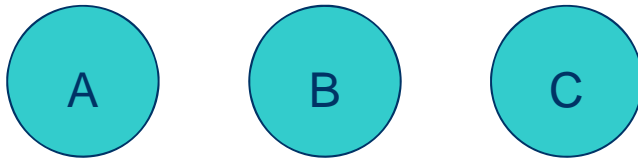
Implementation 1 - array

- Abstract sequence
- Successor - implicit
- Predecessor - implicit



Implementation 2 - array

- Abstract sequence



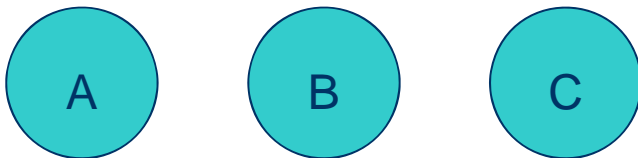
- The reference might be
 - An index into the array
 - A name → lookup

- Successor – explicit
- Predecessor – implicit

node	succ
A	Ref B
B	Ref C
C	α

Implementation 3 - array

- Abstract sequence



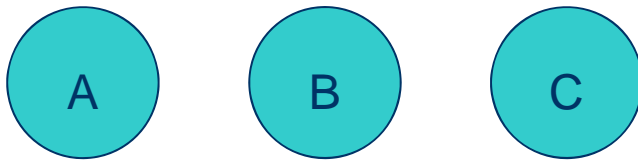
- The reference might be
 - An index into the array
 - A name → lookup

- Successor – explicit
- Predecessor – explicit

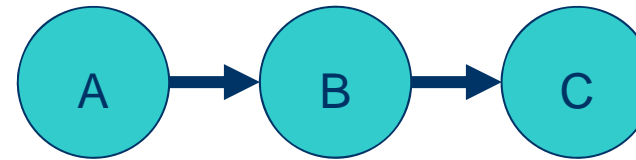
node	succ	pred
A	Ref B	⊥
B	Ref C	Ref A
C	⊥	Ref B

Implementation 4 – singly linked list

- Abstract sequence



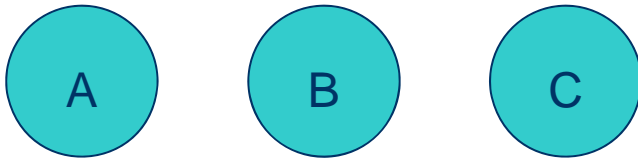
- Successor – explicit
- Predecessor – implicit



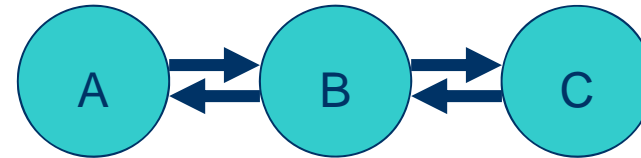
- The arrow is a reference to an element (pointer)

Implementation 5 – doubly linked list

- Abstract sequence



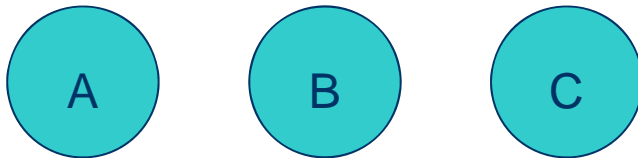
- Successor – explicit
- Predecessor – explicit



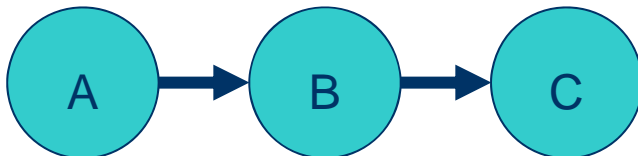
- The arrow is a reference to an element (pointer)

Implementation 6 – directed graph – adjacency list

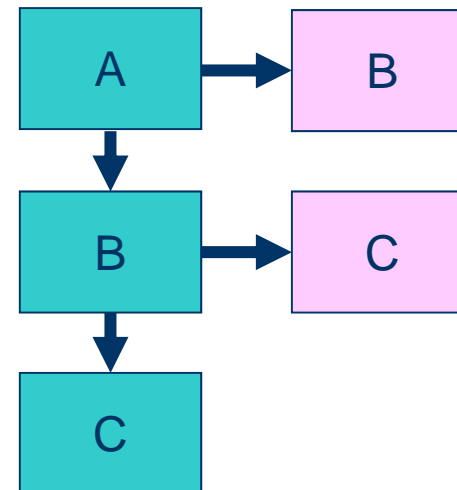
- Abstract sequence



- Graph

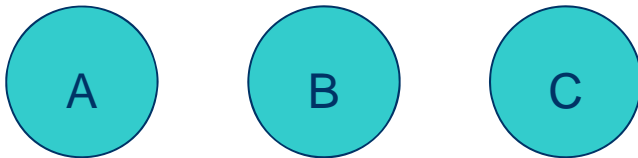


- Successor – explicit
- Predecessor – implicit

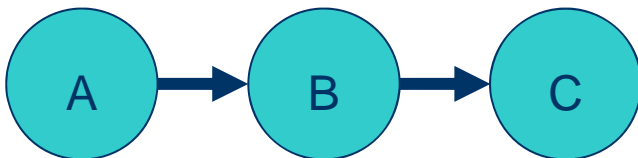


Implementation 7 – directed graph – adjacency matrix

- Abstract sequence



- Graph



- Successor – explicit
- Predecessor – implicit

	A	B	C
A	0	1	0
B	0	0	1
C	0	0	0