1. Array (structure, operations)

1) Representation of polynomial coefficients: not interesting ...

2) Different ways to increase array capacity.

	Copies per Insertion	Unused Memory
Increase by 1	*n* – 1	0
Increase by *m*	*n*/*m*	*m* – 1
Increase by a factor of 2	1	*n*
Increase by a factor of *r* > 1	1/(*r* – 1)	(*r* – 1)*n*

2. 1 Linked List (structure, operations)

1) How to insert a node pointer

2) Given value v and header h, how to find a specific node with value v: only by traversing the linked list.

3) Time complexity:

	Front/1st node	*k*th node	Back/*n*th node
Find	$O(1)$	$O(n)$	$O(1)$
Insert After	$O(1)$	$O(1)$	$O(1)$
Replace	$O(1)$	$O(1)$	$O(1)$
Next	$O(1)$	$O(1)$	n/a
Previous	n/a	$O(n)$	$O(n)$

But students did not study O or Θ representation until lec 3.

4) How to use an array to implement a linked list

2. 2 Special linked list: Doubly Linked List (structure, operations)

1) Similar to Linked list

1.1) benefit: $O(1)$ ***Previous*** .

1.2) drawback: $\Theta(n)$ extra space.

3. Stack (structure, operations)

1) Parsing Reverse-Polish Notation with Stack: e.g. valid notation, calculating results.

2) Check if the given push and pop sequence of Stack is valid or not.

4. Queue (structure, operations)

1. How to use an array to represent a queue.

2. Circular array

3. deque