

Habib University
shaping futures

CS 201 Data Structure II (L2 / L5)

Abstract Data Type

Muhammad Qasim Pasta

qasim.pasta@sse.habib.edu.pk

Slides are designed to be filled during the lectures using board and marks. Some details are not mentioned and these slides should not be used as reading.



Class Activity

We want to create an ADT that store numerical data with the following operations:

- Insert an element into the collection at any specific location
- Remove an element from any specific location of the collection
- Insert/remove an element at/from the beginning of the existing elements in the collection
- Insert/remove an element at/from the last position of the existing elements in the collection
- Returns the sum of all elements in the collection
- Returns the position of an element in the collection
- Returns the position of an element in the sorted collection



USet Interface / ADT

- Represent an unordered set of unique elements

Operations:

- **Add(x)**: add the element if not already exists
- **Remove(x)**: remove x from the set
- **Find(x)**: find x in the set if it exists
- **Size()**: returns the number of elements in the set

For Sset: Sorted Set

- **Find(x)**:
 - find the smallest element y in the set such that $y \geq x$, or returns nil if no such element exists.



ArrayStack: Stack ATD – using Array

- Implementing Stack operations using array
- Array will be used to store the data i.e. elements
- Array has its limitations: homogenous and size (why?)
- Operations:
 - Push(x)
 - Pop()
 - IsEmpty()
 - Size()
- What if we do not want to limit the size – using array?



ArrayQueue: Queue ATD – using Array

- Implementing Queue operations using array
- Array will be used to store the data i.e. elements
- Array has its limitations: homogenous and size (why?)
- Operations:
 - Enqueue(x)
 - Dequeue()
 - IsEmpty()
 - Size()
- Circular array – modulus operator
- What if we do not want to limit the size – using array?



ArrayList: List ATD – using Array

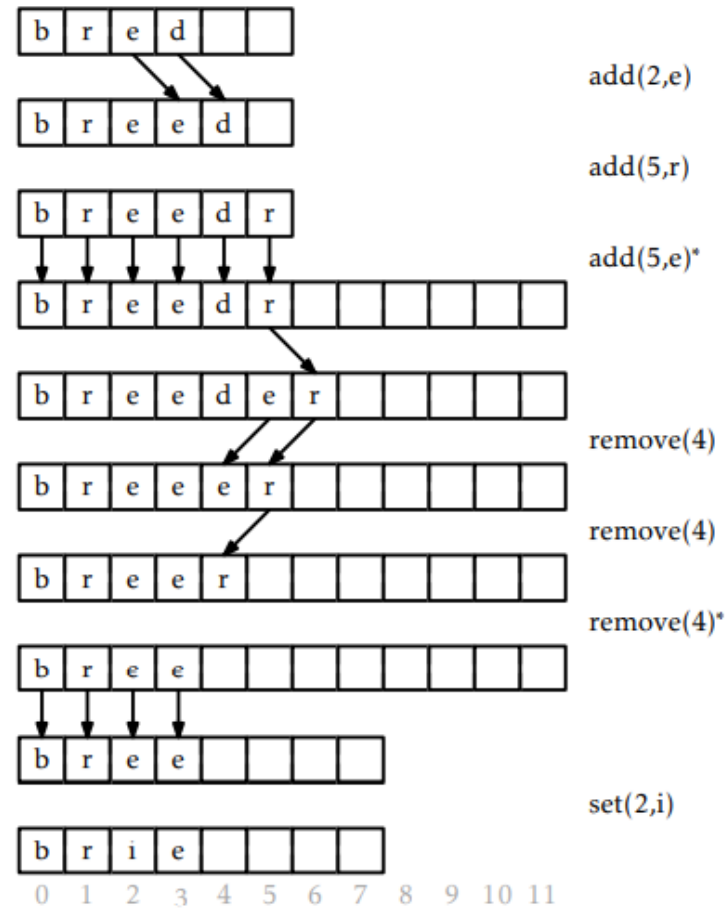
- Implementing List operations using array
- Array will be used to store the data i.e. elements
- Array has its limitations: homogenous and size (why?)
- Operations:
 - get(i)
 - set(i,x)
 - add(i,x)
 - remove(i)
- What if we do not want to limit the size – using array?



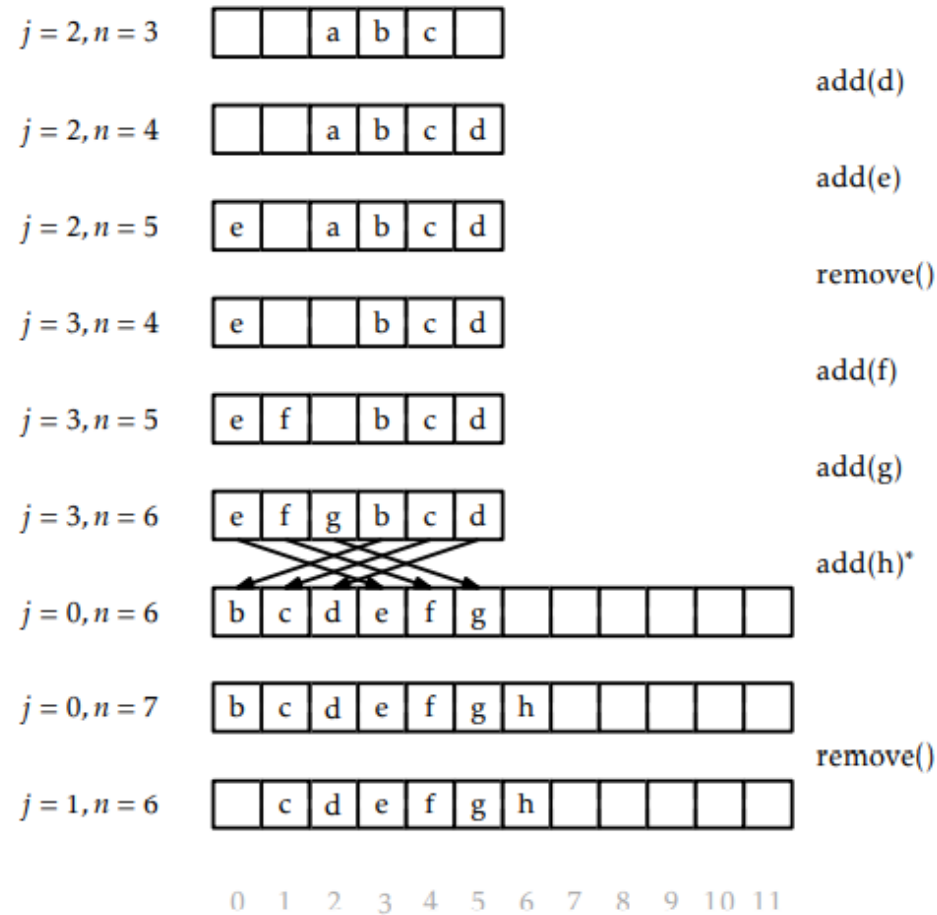
Growing and Shrinking

- Because of fixed size array
 - Scenario when the number of elements can't be determined before hand
 - Either create a enough large array – issues?
 - Recreate array and copy elements – issues?
- Resize the array as per usage
- Grow – if the array is full
 - Increase the size by $2n$
- Shrink – if array size is greater than $3 * n$
 - Decrease the size by $n/2$

List: Add / Remove operations for with resize



Queue: Add / Remove operations with resize





Time Complexity (without resize)

Stack

- Push
- Pop

Queue

- Enqueue
- DeQueue

List

- `get(i)`
- `set(i,x)`
- `remove(i)`
- `add(i, x)`



Reading

- Chapter #2
- Next Class: Amortized Analysis