Cookies and Cram: ITI1121

Introduction to the world of Java

Important Stuff

- Exceptions
- Stack/Queue Architecture
- Linked List / Double Linked Link
- Iterators
- Binary Search Trees

Exceptions

- Throwable Objects that notify the program/programmer of an unexpected/disallowed event
- Java native Exceptions exist, but you can also create your own by extending Exception

```
public class TestException extends Exception {
    public TestException(String message) {
        super(message);
    }
    public TestException(Throwable cause) {
        super(cause);
    }
}
```

Stack

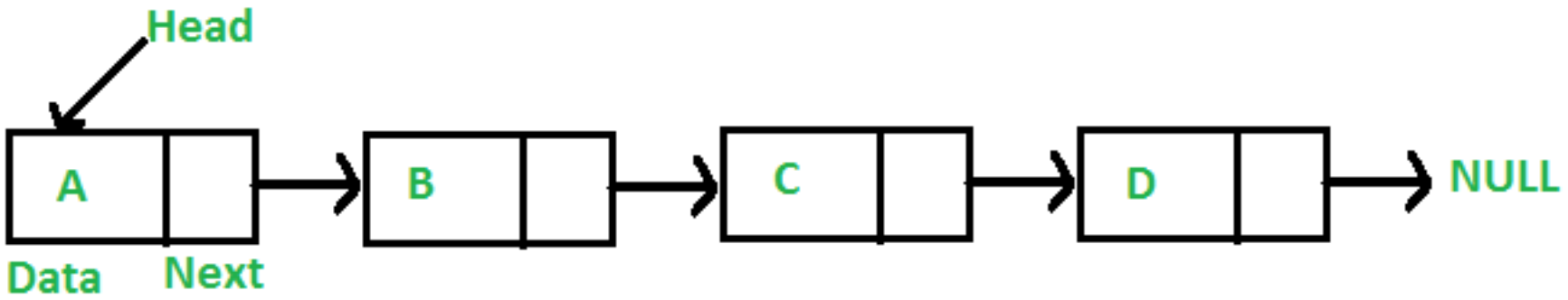
- First in, last out
- Imagine a physical stack of items where you only have access to the top element.
- Used in **many** applications, including hardware, recursion, etc.
- Methods:
  - boolean empty()
  - Element pop()
  - void push(Element element)

Queue

- First in, last out
- Imagine a lineup of people outside of Loblaws (socially distanced, of course)
- Also used in several applications (music queues, queuing network requests)
- Methods:
  - boolean empty()
  - void add(Element element)
  - Element remove()

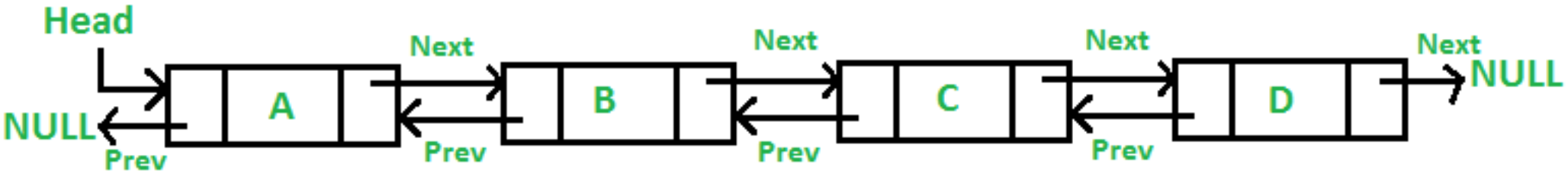
Singly Linked List

- Abstract architecture where elements are connected through pointers
- Only one connection per element (next OR previous element)
- Can be used to implement a queue or a stack
- Advantage over doubly linked list: space



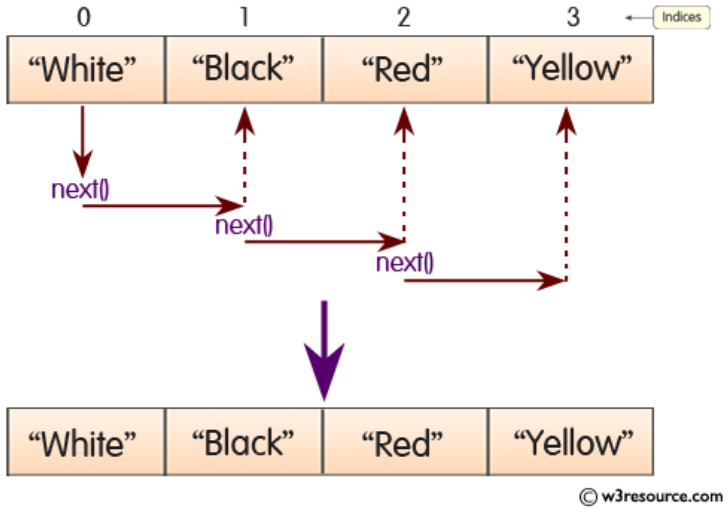
Doubly Linked List

- Abstract architecture where elements are connected through pointers
- Two connections per element (next and previous element)
- Implement binary search trees, SparseVectors (as we will see), Deques, circular lists/circular queues
- Advantage: more knowledge and back-and-forth movement within elements



Iterators

- Custom classes/architectures that help a programmer iterate over your custom architecture with ease.
  - Very generic, anyone can use it without seeing your code
- Two methods:
  - boolean hasNext()
  - Element next()



Binary Trees

- Binary trees are the backbone of several architectures, including but not limited to:
  - Some database structures
  - Heaps
  - Binary search trees
  - P2P programs using has trees
  - Some compression algorithms
- Use recursion for traversal
  - Pre-order
  - In-order
  - Post-order