

Algorithm Sort RR (L).

if L.isEmpty() then

return L.

cur  $\leftarrow$  L.first().

while !L.isLast(cur) do

cur  $\leftarrow$  L.after(cur).

b  $\leftarrow$  cur

while !L.isFirst(b) do {

c  $\leftarrow$  L.before(b).

if c.color = RED  $\wedge$  b.color = BLUE then.

L.swapElement(c, b).

b  $\leftarrow$  c

}



# Algorithm sort RGB (L)

if L.isEmpty() then  
return L

cur ← L.first()

while ! L.isLast(cur) do

cur ← L.after(cur)

b ← ~~cur~~ (cur)

while ! L.isFirst(b)

~~cur~~ ← L.before(b)

if b.color() != c.color() ∧ b.color() = RED then  
L.swapElement(b, c)

if b.color() = BLUE ∧ c.color() = GREEN

L.swapElement(b, c)

b ← c

## Outline and Reading

- The Stack ADT (§2.1)
- Applications of Stacks (§2.1)
- Array-based implementation (§2.1)
- Growable array-based Stack (homework)

## Exceptions

- Operations on the ADT may cause an error condition, an exception.
- Exceptions are said to be "thrown" when an operation cannot be executed.
- Attempting a pop or peek on an empty stack causes an exception to be thrown.
- Operations pop and top cannot be performed if the stack is empty.

## Runtime Stack in the JVM

- The Java Virtual Machine (JVM) keeps track of the chain of active methods with a stack.
- When a method is called, the JVM pushes onto the stack a frame containing:
  - Local variables and return value
  - Program counter, keeping track of the statement being executed
  - When a method ends its frame is popped from the stack and control is passed to the method caller of the stack.
- These are called stack frames or activation records.



~~Handwritten scribbles~~

$\wedge$  and. conjunction

$\vee$  or disjunction

$\cap$  intersection.

$\cup$  union

Algorithm  $\text{noduplicate}(A, B)$

PQ  $\leftarrow$  new priority queue using C.

while ! A.isEmpty() do.

PQ.insert(A.remove(A.first()));

insertLast

while ! B.is Empty() do.

PQ.insertLast(B.remove(B.first()), B).

current  $\leftarrow$  ~~removeMin~~ PQ.removeMin()

while ! PQ.is Empty() do.  $\rightarrow$  A.InsertLast(current)

~~if~~ e  $\leftarrow$  PQ.removeMin()

if current  $\neq$  e then.

A.insertLast(e)

return A

