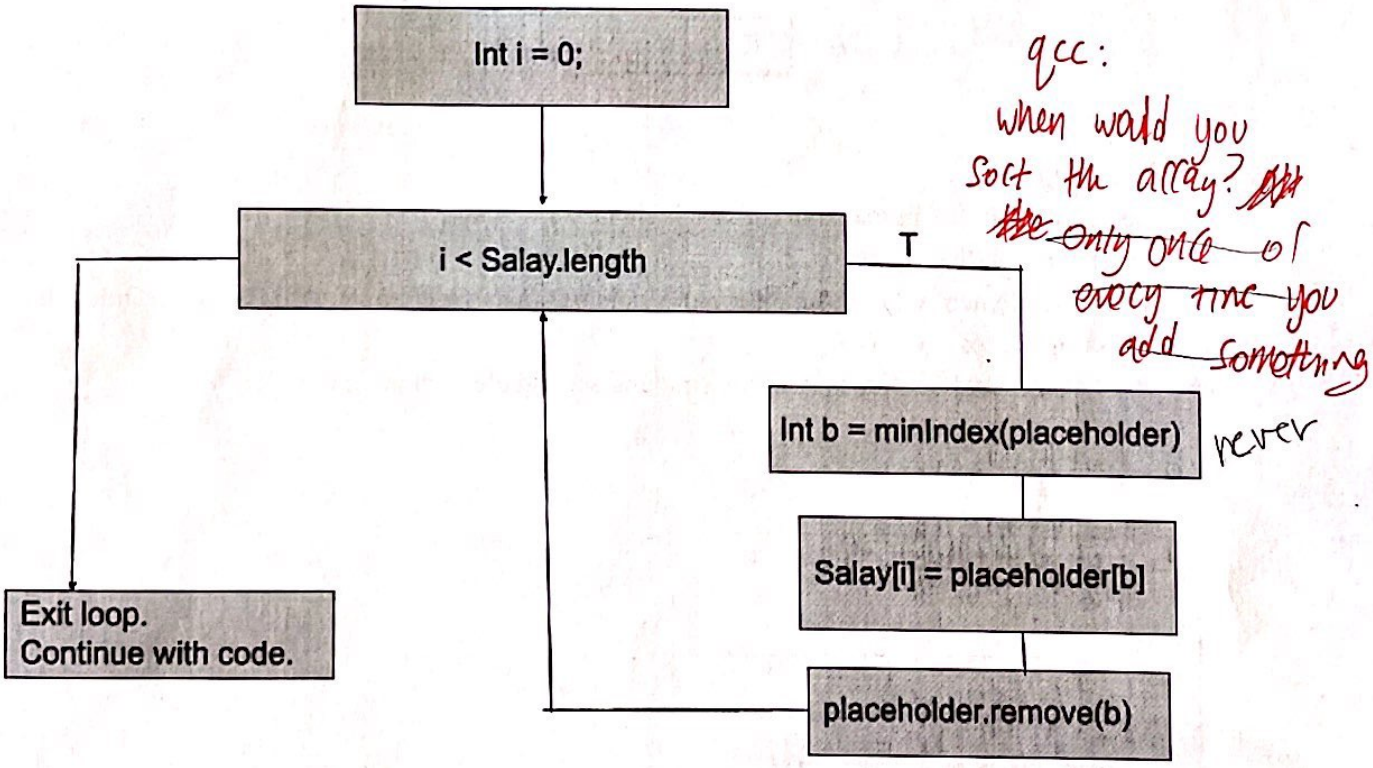
Team Frogs (Ruby Friedman, Ivina Wang, Samantha Hua)
APCS
HW46 -- Wrap the Wrapper
2021-12-09
Time Spent: 1.0 hrs

Reviewed by: SWAG SWASH

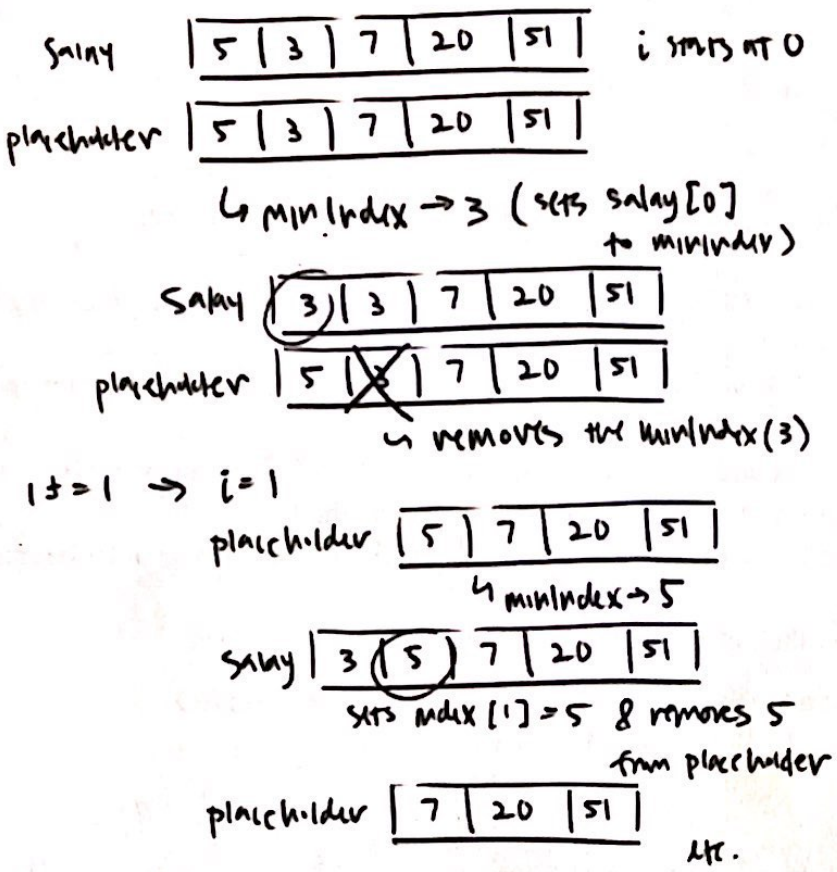
1st strategy:

Start by creating a "placeholder" array that contains all the values of Salay, our instance of SuperArray.
Use a for loop with a increment i that starts at 0 and increases by one for each iteration as long as i is less than Salay.length.
Use a helper function called **minIndex** which uses a for loop to find the index of the minimum value in a given array.
Find the minIndex of placeholder and replace Salay[i] with the value of placeholder[minIndex].
Remove the minIndex from array placeholder using the remove method.
Continue looping through the for loop, now there will be a new minIndex in placeholder because the old value was removed.
Continue to append the new values of min index on to Salay in increasing order.

Placeholder has the same values as the inputted array, but does NOT point to the same array.



qcc:
when would you sort the array? ~~Also~~ ~~Use~~ Only once or every time you add something never

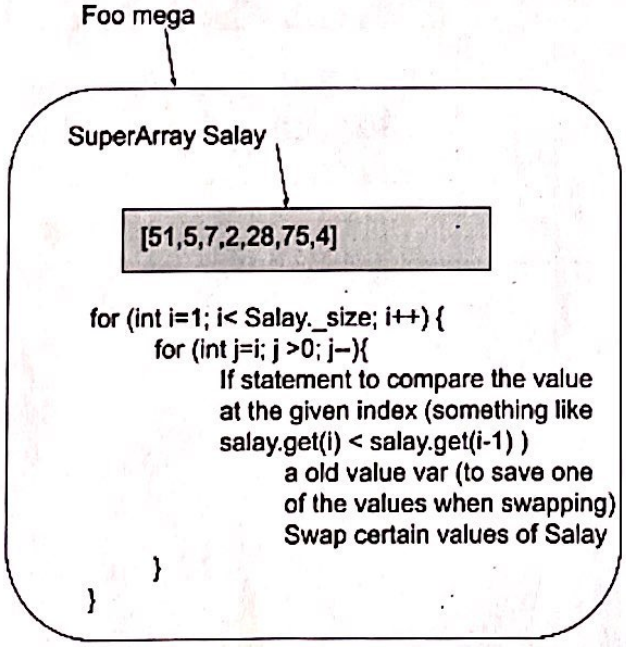


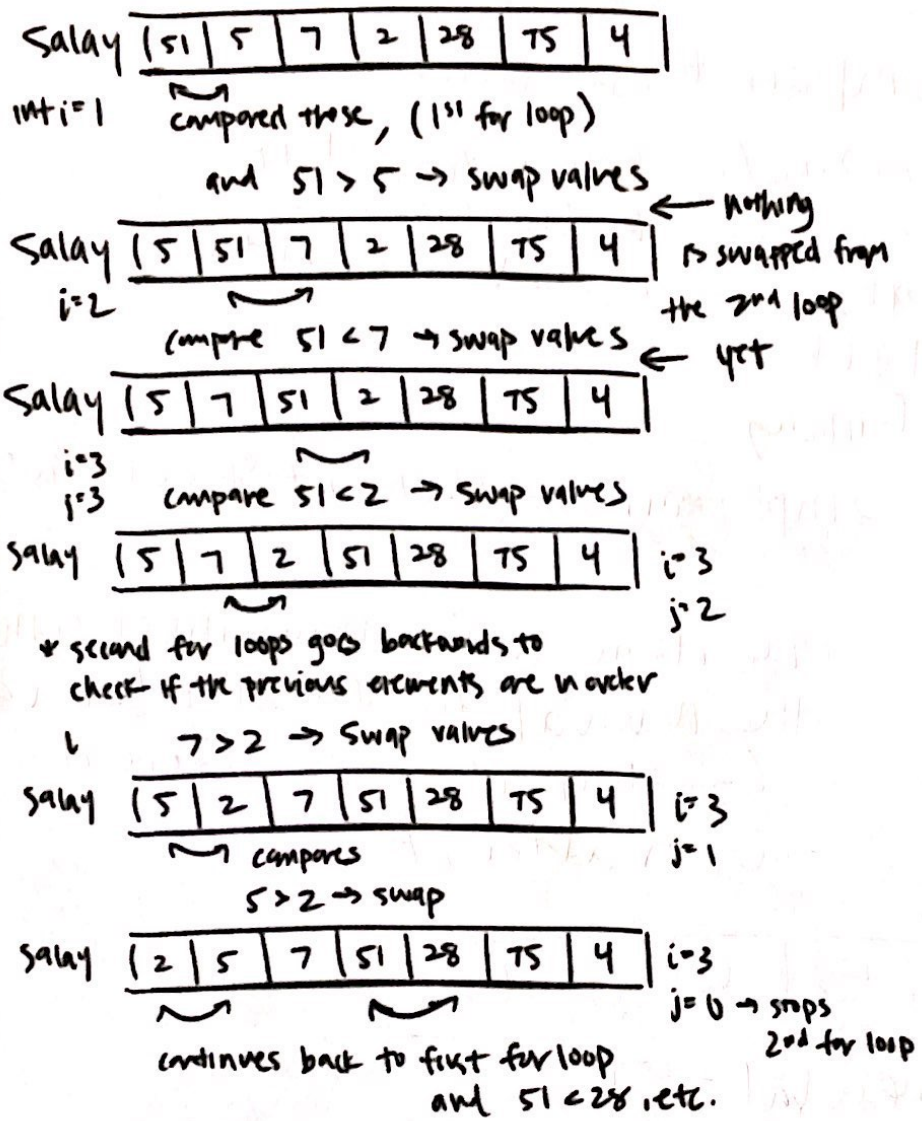
2nd strategy

An outer for loop that iterates through each element in Salay

An inner for loop that compares each element to the one that comes before it

- If the element with the smaller index is larger than the element with the larger index, they swap places
- This process continues until the first and second element are compared





{ }

add(3)

{3}

add(5)

{3,5}

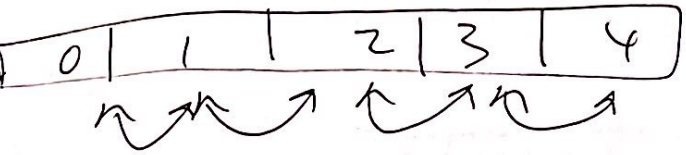
add(4)

{3,4,5}

the sorting should return the index at which the new thing should be added

Sorted or Not

ArrayList<Integer>



Compare these values

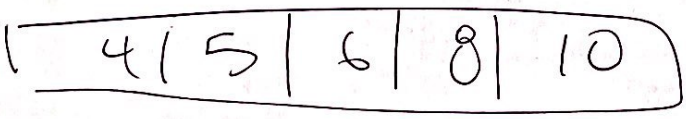
. compareTo will return -1 if the value on the right is larger than the value on the left.

~~add~~ add Binary

ArrayList<Integer>

empty array - element should just be added

one item in the array - must compare the newVal to the singular value
↳ either it goes before it
↳ or after it

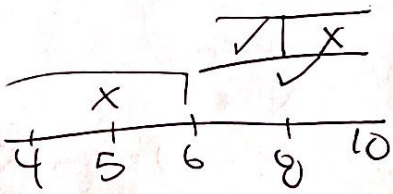


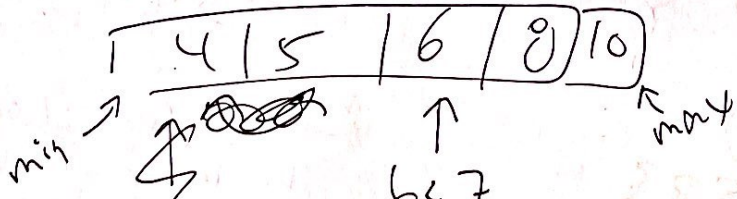
newVal = 7

min = 0

max = 4 → index of last value in the array

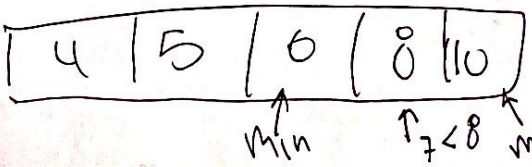
mid = $\frac{(4-0)}{2} = 2$





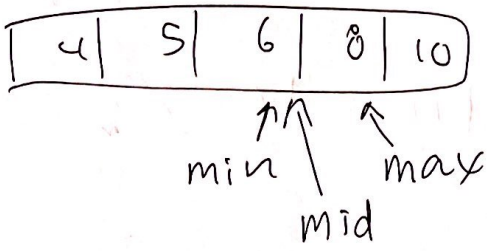
~~4 < 7~~ so we only look at the top half and min becomes mid & +1

then

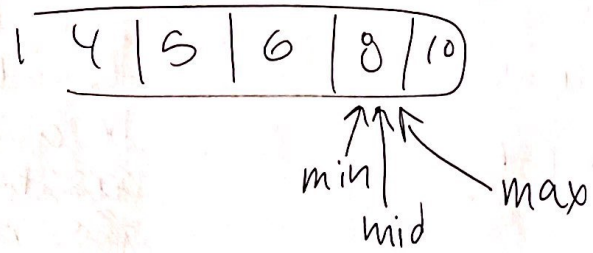


so we decrease the max (max = mid - 1)

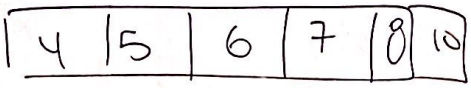
add Binary (cont.)



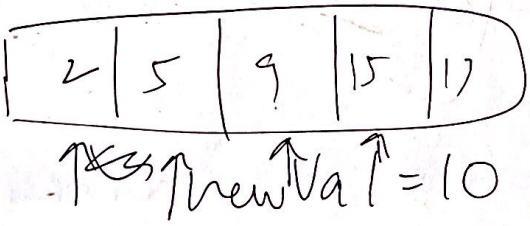
6 < 7 so min = mid + 1



min = max so we add newVal at mid



add Linear



2 < 10
5 < 10
9 < 10
10 < 15 !!! add newVal at i = 2

min = 0

max = 10

mid = (min + max)/2

while {

min ++

max ++

}

mid needs to be updated inside the while loop because ~~[illegible]~~ min + max are looking at the values of those variables.

min = updated

max = updated

a = 5

b = 9

C = a + b

c = 14