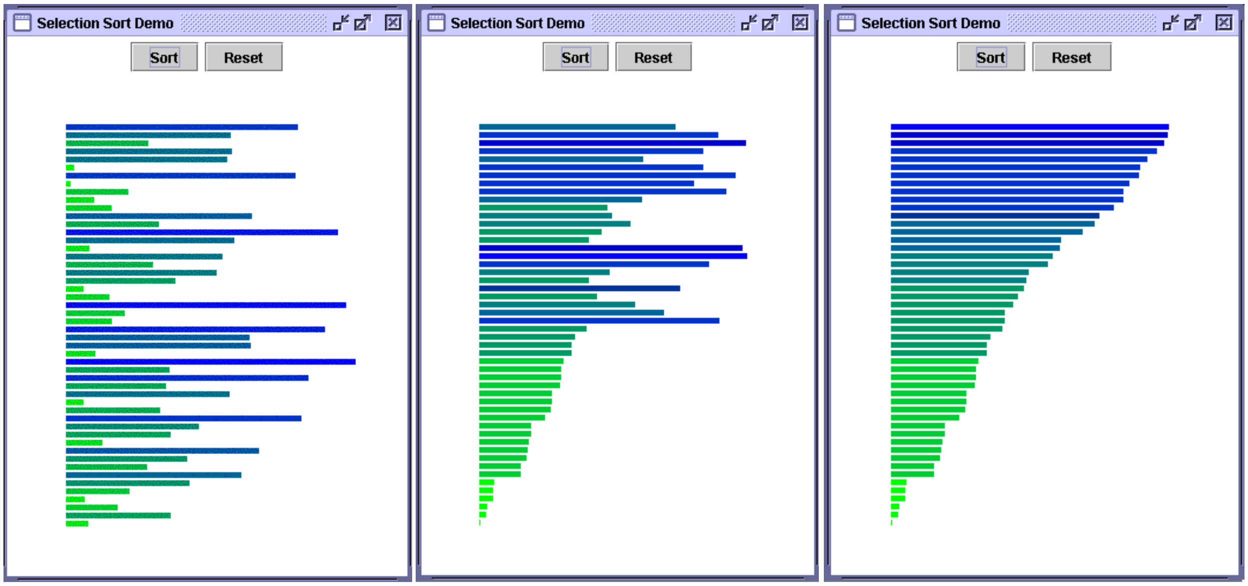
Sorting Animator



- Add in the initial array. It should have 10 values between 1 and 15.
- Code the reset button.

```
import java.awt.*;
import java.applet.*;

public class SortAnimator extends Applet
{
    Button select, bubble, insert;
    int a[] = //add in 10 values. Numbers between 1 and 15
    Button reset;

    public void init ()
    {
        bubble = new Button ("bubble sort");
        add (bubble);
        reset = new Button ("reset");
        add (reset);

    }


    public boolean action (Event e, Object o)
    {
        if (e.target == bubble)
        {
```

```
        bubble (a);
    }
    else if (e.target == reset)
    { //reset all values.
        a [0] = 7;
        printarray (a);

    }
    return true;
}


public void paint (Graphics g)
{
    printarray (a);
}


public void printarray (int a[])
{ //Pre: a is an array with values. It is of size n
    //Post: the values in a are printed to the screen
    Graphics g = getGraphics ();
    g.setColor (Color.white);
    g.fillRect (0, 0, 500, 500);
    int y = 50;

    for (int i = 0 ; i < a.length ; i++)
    {
        g.setColor (new Color (0, a [i] * 15, a [i] * 15));
        g.fillRect (50, y, a [i] * 20, 10);
        y += 12;
    }

}


public void bubble (int a[])
{ //Pre: a is an array with values. It is of size n
    //Post: the values in a are put in ascending order
    int temp;

    for (int i = 0 ; i < a.length - 1 ; i++)
    {
        for (int j = 0 ; j < a.length - 1 - i ; j++)
        { // compare the two neighbours
            if (a [j + 1] < a [j])
```

```
                { //swap the neighbours if necessary
                    temp = a [j];
                    a [j] = a [j + 1];
                    a [j + 1] = temp;
                    for (int m = 0 ; m < 200000000 ; m++)
                        ;
                    printarray (a);
                }
            }
        }
    }
}
```