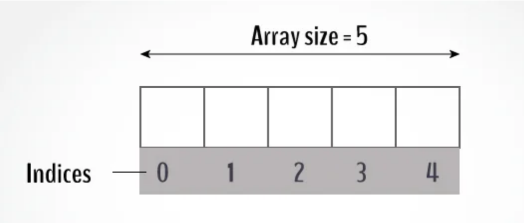
Project: LED Grid - Bubble Sort

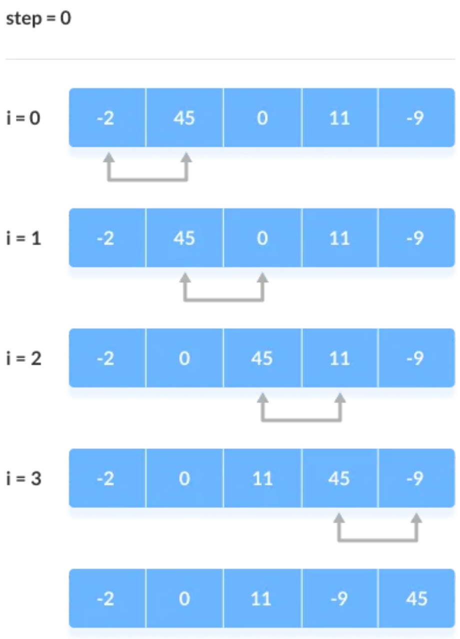
What is an LED matrix?

- Arrays: a data structure made with a collection of items stored contiguously (next to each other)
    - Each item within the array is of the same type (i.e. all items are doubles or Strings)
    - Pro: each item within the collection can be found via an index, which makes items easy to retrieve
        - Indexes **start from 0** and **end with the array size - 1**
    - Con: when arrays are initialized, they are made with a specific size



What is bubble sort?

Bubble sort is a sorting algorithm. Different sorting algorithms have different methods for sorting a list of numbers. Bubble sort's strategy is to continuously iterate through the list to find the largest element and push it towards the end of the list. Essentially, if an element is larger than its neighbor on its right, they switch positions, so the larger one is ahead. The following schematics show a step by step example of bubble sort.



Step 0:

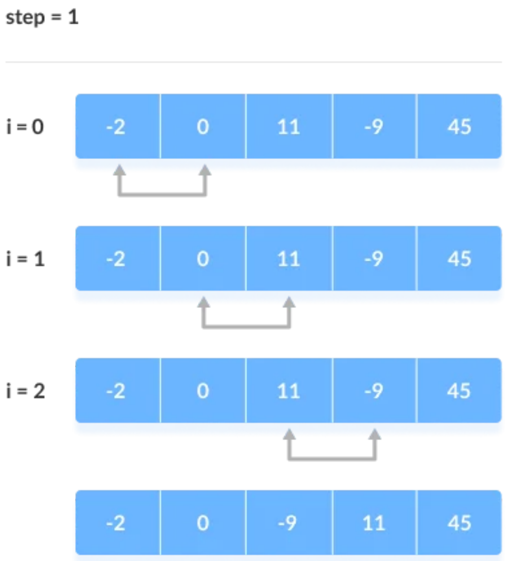
In step 0, our first element is -2.
-2 is NOT greater than its right neighbor, so it stays in the same position.

We move to the next index. 45 is larger than 0, so they switch places.

We move to the next index. 45 is larger than 11 so they switch places.

We move to the next index. 45 is larger than -9, so they switch places.

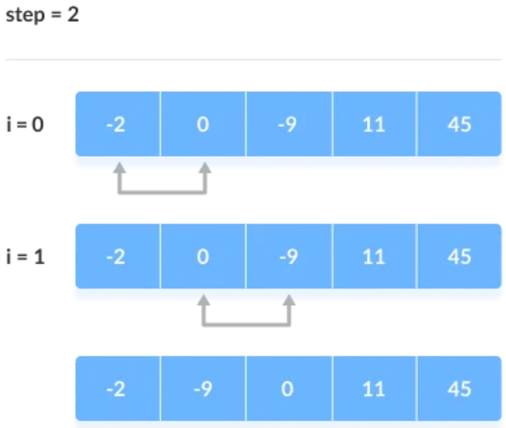
Once we reach the end, we know 45 is the largest element in the list.



Step 1:

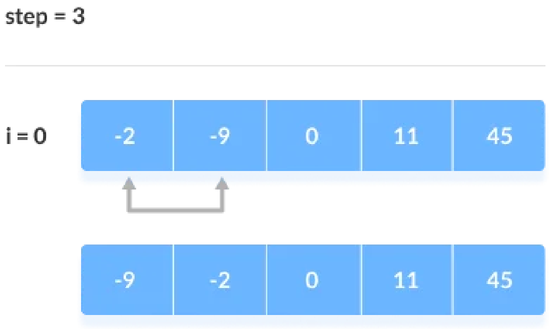
We start the process again, but this time we only need to go up to the second to last index because we know 45 is the largest element already.

Repeat the process from step 0.



Step 2:

Repeat the process of step 0, but only go up to the third to last index.



Step 3:

Now we only have to check the first two elements between each other and put them in their correct locations.

Note that the number of steps taken correlates directly with the number of elements in the list.

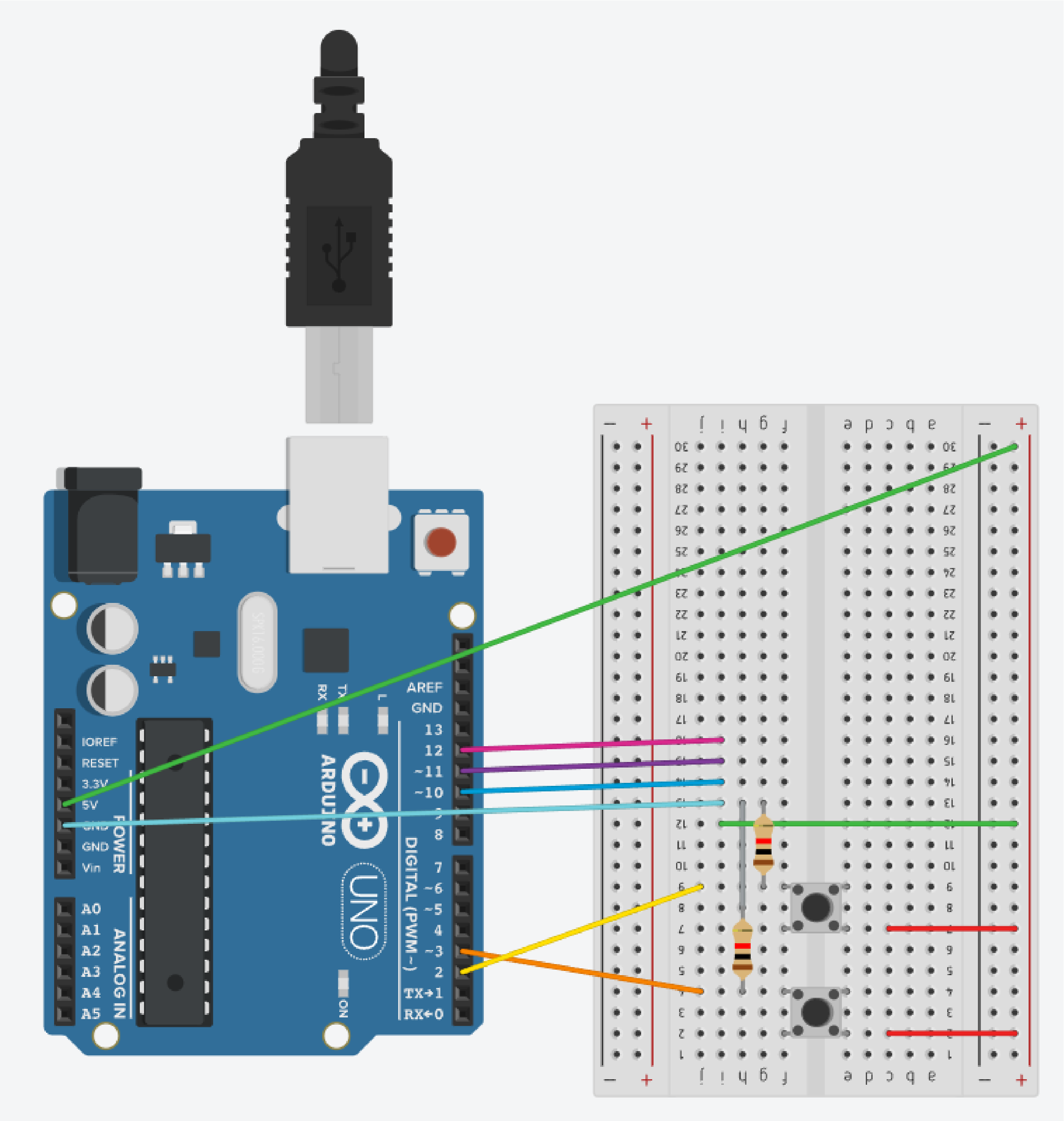
We want to illustrate this sorting method using our LED grids.

Materials Needed:

- 2: Buttons
- 10: Wires
- LED Matrix
- 2: 1k Resistors
- Arduino Uno

Project Demo: LED Grid Bubble Sort

Circuit Diagram:



This circuit diagram does NOT include the LED grid. The LED grid should be placed in row j. With GND (row 13), Vin (5V), DIN (row 14 breadboard), CS (row 15), CLK (row 17)

Code Sample:

```
#include "LedControl.h"

// Set the LED grid pins to specific ports on the Arduino
int DIN = 7;
int CS =  6;
int CLK = 5;

// The button used to generate random numbers
int buttonState = 0;
int buttonPin = 13;

// The button used to sort the numbers
int buttonSortState = 0;
int buttonSortPin = 12;

// The number of rows on the LED grid
int rows = 8;

// The random numbers array used.
int randomNumbers[] = {0,0,0,0,0,0,0,0};

// The LED Matrix object
LedControl lc=LedControl(DIN,CLK,CS,1);
```

```
/**
 * Set up the buttons and matrix. Also sets analog pin 0 to be the pin used
 * to gather random values
 */
void setup() {

  // if analog input pin 0 is unconnected, random analog
  // noise will cause the call to randomSeed() to generate
  // different seed numbers each time the sketch runs.
  // randomSeed() will then shuffle the random function.
  randomSeed(analogRead(0));

  // Sets pins for button inputs
  pinMode(buttonPin, INPUT);
  pinMode(buttonSortPin, INPUT);

   /*
   The MAX72XX is in power-saving mode on startup,
   we have to do a wakeup call. This is mostly important for battery powered
   devices.   */
  lc.shutdown(0,false);

  /* Set the brightness to a medium value (0-15). 0 does not turn off the
     grid */
  lc.setIntensity(0,8);

  /* and clear the display (switches all LEDs to off)*/
  lc.clearDisplay(0);

}

/**
 * Clears the array and displays the empty screen to the LED matrix.
 */
void clearArray(int arr[]){
  // Sets each row of the LED grid to be 0.
   for(int i = 0; i < rows; i++){
    arr[i] = B00000000;
  }

  displayArray(arr);
}
```

```
/**
 * Displays the array of bytes in arr to the LED matrix screen.
 */
void displayArray(int arr[]){
  // Uses x as the base value displayed.
  byte x = B00000001;

  // Sets the bit number from arr[index] to be turned on
  // For example arr[i] = 7 then the 7th LED in the row will be on,
everything else is off.
  for(int i = 0; i < rows; i++){
    int y = x << arr[i];
    lc.setRow(0,i, y);
    delay(20);
  }
}

/**
 * Sorts the array using bubble sort and displays the steps.
 */
void bubbleSort(int arr[]){

  for(int i = 0; i < rows; i++){

    for(int j = 0; j < rows-i - 1; j++){

       if (arr[j] > arr[j+1]) {
        int temp = arr[j];
        arr[j] = arr[j+1];
        arr[j+1] = temp;
      }
      displayArray(arr);
    }

  }

}
```

```
/**
 * Continuously checks for buttons pressed to either generate numbers or sort
the array.
 */
void loop() {
  // put your main code here, to run repeatedly:
  buttonState = digitalRead(buttonPin);
  buttonSortState = digitalRead(buttonSortPin);

  // Generate numbers if the first button pressed
  if(buttonState == HIGH){
    clearArray(randomNumbers);

    // Stores 8 random numbers into an index in the array
    for(int i = 0; i < rows; i++){
      randomNumbers[i] = random(0, 8);
    }

    // Displays the array of random numbers
    displayArray(randomNumbers);
  }

  // Sort the array if the second button pressed
  if(buttonSortState == HIGH){
    bubbleSort(randomNumbers);
  }

}
```