CSE331 HW0

Hao Zhou 2023277

Algorithm

```java
public static void rearrange(int[] b) {
    int l = 0;
    int r = b.length - 1;
    int i = 0
    while (i <= r) {
        if (b[i] < 0) {
            b[l] = b[i];
            i += 1;
            l += 1;
        } else if (b[i] == 0) {
            i += 1;
        } else {
            swap(b[i], b[r]);
            r -= 1;
        }
    }
  for (int i = l; i <= r; i++) {
      b[i] = 0;
  }
}
```

Argument

Every element in the array will be viewed and processed once since we are processing the element i and we will let i++ or r-- until they meet. Every element which is smaller than 0 will be placed from the left and greater than 0 will be placed from the right in each iteration. After all the elements are processed, negative are on the left and positive on the right. Then I let every element between them to be 0.

To be more specific, for every j < l, we have b[j] < 0. For each j > r, we have b[r] > 0. We ensure l <= i and r >= i before the iteration ends. l - 1 == # of negative and length - r == # of positive and r - l + 1 == # of zeros.