Problem 4:

- The algorithm only contains one *for loop* with 4 if-else statements so the time complexity for the loop would *be O(N)* because the run time increases by one as each if or else if statement is added. In the best case the loop would run to its entirety while only comparing to one *if statement*. In the worst case it would run the loop for its entirety and would compare every single *if-else statement*.

```
        arraySize = a.length;

        for (int i = 0; i < arraySize; i += 2) {
            if (a[i] <= a[i+1] && a[i] >= a[i+2]) {
                temp1 = a[i+1];
                a[i+1] = a[i+2];
                a[i+2] = temp1;
            }
            else if (a[i] <= a[i+1] && a[i] <= a[i+2]) {
                temp2 = a[i];
                a[i] = a[i+1];
                a[i+1] = temp2;
            }
            else if(a[i] >= a[i+1] && a[i] >= a[i+2] && a[i+1] >= a[i+2]) {
                temp3 = a[i+1];
                a[i+1] = a[i+2];
                a[i+2] = temp3;
            }
            else {
                a[i] = a[i];
                a[i+1] = a[i+1];
                a[i+2] = a[i+2];
            }
        }
    }

    public static void main (String [] args) {
        int [] a = new int [] {8, 9, 0, 3, 1};
        problem4A(a, 5);
    }
}
```

- There is a linear time solution for this problem because every time another *if statement* is added so that means that the run time is linear. As one increases the other increases as well.