Ronald Chatelier

1(a).

```
for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            for (int k = 1; k < n; k++) {
                std::cout << "test 1";
            }
        }
    } // run 3x^3 [n^3]
    for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            k = n;
            while (k > 0) {
                k = k / 2;
            }
        }
    } // (x^2)logx [(n^2)logn]
    for (int i = 1; i <= 2; i++) {
        for (int j = 1; j < n; j++) {
            for (int k = 1; k < n; k++) {
                std::cout << "test 2";
            }
        }
    } // 2x^2 [2n^2]
    for (int i = 1; i <= 3; i++) {
        k = n;
        while (k > 0)
            k = k / 2;
    } // 3logx [3logn]
```

1 (b).

```
for (int i = 1; i <= 2; i++) {
        for (int j = 1; j < n; j++) {
```

```
                    for (int k = 1; k < n; k++) {
                        std::cout << "test 2");
                    }
            }
    } // 2x^2 [2n^2]
    for (int i = 1; i < n; i++) {
            for (int j = 1; j < n; j++) {
                    k = n;
                    while (k > 0) {
                        k = k / 2;
                    }
            }
    } // (x^2)logx [(n^2)logn]

    for (int i = 1; i <= 3; i++) {
            k = n;
            while (k > 0)
                    k = k / 2;
    } // 3logx [3logn]
```

2.

```
smallestNumber ( array = [a1, a2, …], length(array) = 4)

small_1 = a1

small_2 = 0

for i = 2 to n

if small_1 > small_2

small_1 = small_2

return small_1, small_2

        small_1 = array[i]

        else if small_2 > array[i]
```

3(a).

2 (n - 1 - 1) + 2(n - 1 - 2) + 2(n - 1 - 3) + ...+2(n - 1 - (n - 1)

2 * (n *(n - 1) - 1 * (n - 1) - (n * (n-1) / 2))

2 * (n * (n - 1) /2 - (n - 1)

(n^2) - n - 2n + 2

Time complexity: O(n^2)

3(b).

(n -1) * (m + 1) * t)

(n * m * t)

Time complexity: O(nmt)

4(a).

array [n] [a1…an]

numberPositive = 0

numberNegative = 0

for int i = 0 to n - 1

if array[i] < 0

numberNegative++

else if array[i] > 0

numberPositive++

if numberPositive < numberNegative

return numberNegative

else

return numberPositive

4(b).

power2 = 0

power1 = o

for int i = 0 to n - 1

power1 != power2

return power1, power2

else

return 1;

5.

323 = 124 * 2 + 75

124 = 75 * 1 + 49

75 = 49 * 1 + 26

49 = 26 * 1 + 23

26 = 23 + 3

23 = 3 * 7 + 2

3 = 2 * 1 + 1

2 = 1 * 2 + 0

GCD: 1

1 = 323(s) + 124(t)

s: 43

t: -112