**Tyler Brent**

**William Almond**

**Aaron Chau**

**4.1**

N = The size of the field

M = The number of rocks

S[S1 S2 … Sn] = The location of the rocks

Field-And-Stones(N, M, S)

Count = 0

Max = 0

for I ⟵ 0 to N

for J⟵0 to N

while true

if Matrix[ I + Count][ J + Count ] != 0

&& Matrix[ I + Count][ J + Count +1 ] != 0

&& Matrix[ I + Count + 1 ][ J + Count] != 0

&& Matrix[ I + Count +1 ][ J + Count + 1 ] != 0

Count ⟵ Count + 1

If Count > Max

Max ⟵ Count

Count = 0

Return Max

-END PROGRAM-

Time Complexity of this algorithm is O(n^3)

Each loop is highlighted in yellow.

**4.2**

N = Size of the field

M = Number of stones

S =

Count = Current count of surrounding tiles

Max = Maximum square that can be created

Stones and Fields(N, S)

Max = 0

Count = 0

For

For

If Matrix[ I ] [ j ] is 0

Do nothing

Else if j or I is 0

Matrix[ I ][ j ] ⟵ 1

Else

Count = Matrix[ I ][ j ]

if Matrix[ I + 1][ j ] < count

Count ⟵ Matrix[I + 1][ j ]

if Martrix[ I + 1][ j + 1 ] < Count

Count ⟵ Matrix[ I + 1 ][ j + 1 ]

Count ⟵ Count + 1

Matrix[ I ][ j ] ⟵ Count

If Count > Max

Max ⟵ Count

Return max

-END PROGRAM

Time Complexity of this algorithm is O(n^2)

Each loop is highlighted in yellow.

**4.3**

**(SEE PROGRAM: StonesAndFields.java)**