**Execution Time Serial: 1.125 seconds.**

[40000000, 40000000, 40000000, 40000000, 40000000, 40000000, 40000000, 40000000, 40000000, 40000000]

**Execution Time Version 1: 5.257 seconds.**

This version take more execution time because there are maximum lock than other version because of the less max value.

[40000000, 40000000, 40000000, 40000000, 40000000, 40000000, 40000000, 40000000, 40000000, 40000000]

**Execution Time Version 2: 4.074 seconds.**

This version took less execution time because the max value is greater than the version 1 which is 100. Which mean there are less lock in the loop.

[4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000]

**Execution Time Version 3: 1.531 seconds.**

In this version the output value is wrong because we implement normal int variable for the hist. The execution time is faster than version 2 because there won’t any atomic lock in the loop. Version 3 give us the wrong output because of the data race.

[3898514, 3955497, 3939261, 3944353, 3947250, 3956094, 3955211, 3957662, 3957112, 3957798, 3958883, 3958687, 3959695, 3961638, 3962084, 3962213, 3933374, 3958731, 3950616, 3954873, 3957803, 3963084, 3963192, 3963795, 3962386, 3961372, 3961622, 3961622, 3961994, 3963300, 3963852, 3964588, 3914019, 3963713, 3948219, 3950253, 3950786, 3955623, 3953116, 3951166, 3948139, 3946724, 3947346, 3947087, 3946379, 3948043, 3948741, 3949078, 3908918, 3948082, 3935557, 3940721, 3944154, 3949167, 3949270, 3949159, 3946711, 3946504, 3947523, 3947903, 3948111, 3950052, 3951489, 3952467, 3890014, 3951503, 3933606, 3936025, 3936699, 3941317, 3938475, 3935539, 3930430, 3930179, 3932681, 3933170, 3933270, 3935020, 3935537, 3936490, 3881394, 3934578, 3918375, 3924398, 3928915, 3934175, 3933706, 3933104, 3928925, 3929129, 3931047, 3931424, 3931907, 3934452, 3936098, 3936913, 3919123, 3952734, 3943346, 3949291]

**Execution Time Version 4: 0.565 seconds.**

This version has a least execution time because of the max is 100 and we implement ordinary variable and a global hist array so that there is not any lock in the loop while assigning value of histArray to hist.

[4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000, 4000000]