CBC Latches, Service Time, and Response Time

Author: Craig Shallahamer (craig@orapub.com), Version 1a, 23-Aug-2011

Background and Purpose

The purpose of this notepad is to see if increasing CBC latches can decrease CPU and non-idle wait time per buffer get. And also to see the relationship between the change in the time to process a buffer get and buffer get dependant SQL.

Experimental Data

Below is all the experimental data. The experiment was run on a Dell single four-core CPU, Oracle 11.2G. According to "cat /proc/version": Linux version 2.6.18-164.el5PAE (mockbuild@ca-build10.us.oracle.com) (gcc version 4.1.2 20080704 (Red Hat 4.1.2-46)) #1 SMP Thu Sep 3 02:28:20 EDT 2009.

I created a massive cpu bottleneck by having a number of Oracle sessions run SQL that is logical IO (v$sysstat: session logical reads) dependent. I gathered 30 ten minute samples with the CBC latches set to 256 and then to 32768. During these ten minute collection periods, I am sampling the elapsed time of a specific LIO dependent SQL statement. With 256 CBC latches, around 23 elapsed times where sampled. With 32768 CBC latches, around 71 elapsed times where sampled. The difference in samples was the result of the SQL completing sooner when there was 32768 CBC latches.

The order of sample data is sample number, elapsed time (seconds), buffer gets (session logical reads), instance non-idle wait time (sec), instance CPU consumption (sec), and DB time (sec).

In[1]:=

```
ss256Elapsed = {1, 1, 13.695948, 1, 2, 20.172087, 1, 3, 19.662529, 1, 4, 23.66364, 1, 5, 18.479148, 1, 6,
    18.856306, 1, 7, 20.440646, 1, 8, 21.990721, 1, 9, 19.084604, 1, 10, 22.506497, 1, 11, 21.100975,
    1, 12, 17.469718, 1, 13, 20.614966, 1, 14, 23.673176, 1, 15, 20.95516, 1, 16, 18.447856, 1,
    17, 21.590476, 1, 18, 21.212463, 1, 19, 22.331199, 1, 20, 19.112634, 1, 21, 22.121533, 1, 22,
    16.846098, 1, 23, 16.903432, 2, 1, 20.117696, 2, 2, 21.725682, 2, 3, 21.831289, 2, 4, 18.741399, 2,
    5, 21.021116, 2, 6, 18.52062, 2, 7, 18.93537, 2, 8, 21.477285, 2, 9, 16.319703, 2, 10, 21.027473,
    2, 11, 20.658319, 2, 12, 21.773501, 2, 13, 21.312561, 2, 14, 15.686597, 2, 15, 22.245191, 2,
    16, 16.810937, 2, 17, 17.730489, 2, 18, 22.643744, 2, 19, 23.109687, 2, 20, 21.755056, 2, 21,
    20.854011, 2, 22, 24.453551, 2, 23, 21.367241, 3, 1, 3.244524, 3, 2, 21.911371, 3, 3, 24.580335, 3,
    4, 20.568654, 3, 5, 17.677306, 3, 6, 18.099442, 3, 7, 21.430922, 3, 8, 21.499659, 3, 9, 18.46176,
    3, 10, 15.396508, 3, 11, 21.040791, 3, 12, 23.902254, 3, 13, 24.326798, 3, 14, 21.047732, 3,
    15, 20.247645, 3, 16, 21.775596, 3, 17, 16.948469, 3, 18, 23.199472, 3, 19, 24.547069, 3, 20,
    17.709302, 3, 21, 20.043618, 3, 22, 18.41053, 3, 23, 22.75477, 4, 1, 18.855299, 4, 2, 22.520426, 4,
    3, 20.887656, 4, 4, 17.155155, 4, 5, 23.50172, 4, 6, 18.783711, 4, 7, 19.683887, 4, 8, 23.408615,
    4, 9, 19.220871, 4, 10, 18.91464, 4, 11, 22.610806, 4, 12, 20.569813, 4, 13, 19.83096, 4, 14,
    18.131507, 4, 15, 16.076485, 4, 16, 24.214894, 4, 17, 23.227945, 4, 18, 20.720694, 4, 19, 18.066112,
    4, 20, 20.279586, 4, 21, 24.815986, 4, 22, 17.305528, 4, 23, 21.766586, 5, 1, 4.70212, 5, 2,
    23.528703, 5, 3, 22.255673, 5, 4, 17.253146, 5, 5, 21.995851, 5, 6, 21.854606, 5, 7, 18.915813,
    5, 8, 20.608713, 5, 9, 23.22468, 5, 10, 19.007827, 5, 11, 20.281022, 5, 12, 21.877953, 5, 13,
    20.13709, 5, 14, 18.26219, 5, 15, 23.119477, 5, 16, 20.220831, 5, 17, 21.55273, 5, 18, 21.087662,
    5, 19, 18.532178, 5, 20, 24.680503, 5, 21, 21.768813, 5, 22, 21.885742, 5, 23, 19.008701, 6, 1,
    1.524795, 6, 2, 21.438191, 6, 3, 24.153896, 6, 4, 21.045452, 6, 5, 16.843272, 6, 6, 22.431163,
    6, 7, 19.422878, 6, 8, 20.182342, 6, 9, 19.33126, 6, 10, 19.414659, 6, 11, 22.373934, 6, 12,
    18.460977, 6, 13, 20.510164, 6, 14, 22.916455, 6, 15, 20.79946, 6, 16, 24.817833, 6, 17, 19.827911,
    6, 18, 18.446284, 6, 19, 20.111823, 6, 20, 17.136381, 6, 21, 21.440676, 6, 22, 21.933965, 6, 23,
    24.591259, 7, 1, 21.256177, 7, 2, 21.89176, 7, 3, 17.254656, 7, 4, 23.643945, 7, 5, 23.213842, 7,
    6, 22.569886, 7, 7, 14.29188, 7, 8, 20.278627, 7, 9, 15.479665, 7, 10, 23.610064, 7, 11, 19.453619,
    7, 12, 20.889812, 7, 13, 24.021473, 7, 14, 24.418039, 7, 15, 20.317918, 7, 16, 19.787339, 7,
    17, 19.002069, 7, 18, 20.103889, 7, 19, 19.821766, 7, 20, 15.501163, 7, 21, 21.690752, 7, 22,
    24.048086, 7, 23, 22.662453, 8, 1, 11.681827, 8, 2, 21.494788, 8, 3, 21.28822, 8, 4, 22.785768, 8,
    5, 23.10174, 8, 6, 21.584343, 8, 7, 21.899641, 8, 8, 19.205554, 8, 9, 18.07786, 8, 10, 20.272283,
    8, 11, 22.134854, 8, 12, 24.894011, 8, 13, 21.017717, 8, 14, 21.646814, 8, 15, 19.453715, 8, 16,
    23.562715, 8, 17, 23.141919, 8, 18, 21.934588, 8, 19, 20.14671, 8, 20, 23.56353, 8, 21, 20.744869,
    8, 22, 24.032751, 9, 1, 22.059643, 9, 2, 21.370142, 9, 3, 19.463205, 9, 4, 21.930264, 9, 5,
    21.606932, 9, 6, 15.006787, 9, 7, 24.027577, 9, 8, 21.466864, 9, 9, 22.271641, 9, 10, 15.046097,
```

22.328722, 9, 22, 23.303762, 9, 23, 23.314906, 10, 1, 4.661392, 10, 2, 23.237348, 10, 3, 23.46569,
10, 4, 21.050194, 10, 5, 22.557133, 10, 6, 17.247327, 10, 7, 20.226941, 10, 8, 21.184723, 10, 9,
20.313144, 10, 10, 21.066553, 10, 11, 22.060395, 10, 12, 22.316539, 10, 13, 23.232979, 10, 14,
18.931062, 10, 15, 17.208819, 10, 16, 20.713781, 10, 17, 22.213656, 10, 18, 21.939829, 10, 19,
19.780565, 10, 20, 21.671578, 10, 21, 22.677605, 10, 22, 21.519857, 10, 23, 23.478499, 11, 1,
8.014086, 11, 2, 23.569373, 11, 3, 18.961973, 11, 4, 22.03431, 11, 5, 25.2788, 11, 6, 17.674027,
11, 7, 19.385042, 11, 8, 21.867738, 11, 9, 20.599039, 11, 10, 22.817782, 11, 11, 20.923642,
11, 12, 20.624734, 11, 13, 23.202691, 11, 14, 22.29046, 11, 15, 21.503308, 11, 16, 19.907742,
11, 17, 22.319666, 11, 18, 21.541889, 11, 19, 16.424855, 11, 20, 22.834504, 11, 21, 22.732602,
11, 22, 20.218971, 11, 23, 21.063789, 12, 1, 9.811638, 12, 2, 22.79863, 12, 3, 21.027223, 12,
4, 23.004588, 12, 5, 19.789845, 12, 6, 19.102888, 12, 7, 24.531837, 12, 8, 17.605376, 12, 9,
23.304581, 12, 10, 19.345429, 12, 11, 23.26257, 12, 12, 22.448159, 12, 13, 21.470756, 12, 14,
24.514044, 12, 15, 18.89484, 12, 16, 18.07167, 12, 17, 21.567925, 12, 18, 16.679027, 12, 19,
21.463824, 12, 20, 19.225085, 12, 21, 20.32408, 12, 22, 23.126818, 12, 23, 19.855422, 13, 1,
4.553196, 13, 2, 19.409674, 13, 3, 21.05842, 13, 4, 21.534767, 13, 5, 21.865394, 13, 6, 21.113863,
13, 7, 21.477739, 13, 8, 20.618332, 13, 9, 17.192721, 13, 10, 21.191911, 13, 11, 22.893738,
13, 12, 21.129821, 13, 13, 20.80576, 13, 14, 18.935087, 13, 15, 21.341857, 13, 16, 21.911783,
13, 17, 24.602783, 13, 18, 19.774006, 13, 19, 19.873233, 13, 20, 19.811092, 13, 21, 20.0339,
13, 22, 18.927112, 13, 23, 18.388821, 13, 24, 20.316346, 14, 1, 19.588424, 14, 2, 21.638116,
14, 3, 21.875822, 14, 4, 22.771804, 14, 5, 18.332143, 14, 6, 20.228008, 14, 7, 16.862913, 14,
8, 23.10515, 14, 9, 24.355199, 14, 10, 22.864302, 14, 11, 21.075801, 14, 12, 21.609665, 14,
13, 22.051661, 14, 14, 20.697376, 14, 15, 17.151788, 14, 16, 20.610817, 14, 17, 23.731772, 14,
18, 21.400098, 14, 19, 18.786002, 14, 20, 19.447591, 14, 21, 23.304864, 14, 22, 23.441638,
14, 23, 20.296875, 15, 1, 18.922184, 15, 2, 21.646802, 15, 3, 23.697409, 15, 4, 26.900576, 15,
5, 29.1486, 15, 6, 29.374845, 15, 7, 25.095102, 15, 8, 21.215167, 15, 9, 25.456766, 15, 10,
27.861895, 15, 11, 28.165668, 15, 12, 25.269057, 15, 13, 26.665866, 15, 14, 26.720072, 15, 15,
24.999858, 15, 16, 28.193904, 15, 17, 24.155136, 15, 18, 27.587056, 15, 19, 27.064508, 16, 1,
3.34172, 16, 2, 27.747407, 16, 3, 29.822387, 16, 4, 25.668648, 16, 5, 28.533229, 16, 6, 25.873554,
16, 7, 26.792914, 16, 8, 23.153371, 16, 9, 22.332008, 16, 10, 26.787881, 16, 11, 28.000372,
16, 12, 27.340834, 16, 13, 27.197737, 16, 14, 26.766888, 16, 15, 24.497565, 16, 16, 26.291744,
16, 17, 24.355984, 16, 18, 28.413082, 16, 19, 29.062764, 17, 1, 29.31262, 17, 2, 21.173687,
17, 3, 26.473356, 17, 4, 24.109662, 17, 5, 27.751844, 17, 6, 25.438615, 17, 7, 27.29345, 17,
8, 27.38901, 17, 9, 22.350797, 17, 10, 27.621468, 17, 11, 28.145815, 17, 12, 24.560114, 17,
13, 26.432397, 17, 14, 27.049308, 17, 15, 21.067991, 17, 16, 26.685643, 17, 17, 28.818408, 17,
18, 27.024694, 17, 19, 21.932866, 18, 1, 8.161055, 18, 2, 29.685747, 18, 3, 22.40155, 18, 4,
25.657419, 18, 5, 27.7904, 18, 6, 29.617744, 18, 7, 25.967928, 18, 8, 26.127116, 18, 9, 27.657838,
18, 10, 22.056704, 18, 11, 22.95967, 18, 12, 24.456394, 18, 13, 27.784061, 18, 14, 25.935759,
18, 15, 26.36066, 18, 16, 22.553437, 18, 17, 26.049771, 18, 18, 27.391763, 18, 19, 25.161139,
18, 20, 27.706295, 19, 1, 27.515255, 19, 2, 27.700218, 19, 3, 22.097951, 19, 4, 27.563022,
19, 5, 30.361474, 19, 6, 29.535284, 19, 7, 27.982448, 19, 8, 20.264982, 19, 9, 29.150445, 19,
10, 25.889173, 19, 11, 26.891374, 19, 12, 22.825902, 19, 13, 25.619866, 19, 14, 28.1856, 19,
15, 26.571653, 19, 16, 26.071574, 19, 17, 25.696823, 19, 18, 22.398779, 19, 19, 27.223887,
20, 1, 20.667012, 20, 2, 26.59899, 20, 3, 24.322861, 20, 4, 26.489631, 20, 5, 24.734575, 20,
6, 25.250423, 20, 7, 29.200477, 20, 8, 23.248017, 20, 9, 26.061988, 20, 10, 23.29757, 20, 11,
25.251893, 20, 12, 20.740213, 20, 13, 27.525056, 20, 14, 28.420017, 20, 15, 24.856307, 20,
16, 29.109242, 20, 17, 30.148789, 20, 18, 27.316712, 20, 19, 21.906114, 21, 1, 23.827348, 21,
2, 24.287591, 21, 3, 25.266664, 21, 4, 28.752454, 21, 5, 28.342624, 21, 6, 26.132125, 21, 7,
22.363317, 21, 8, 28.741554, 21, 9, 25.945327, 21, 10, 23.084239, 21, 11, 27.698444, 21, 12,
27.224888, 21, 13, 21.445128, 21, 14, 29.094173, 21, 15, 23.886909, 21, 16, 25.593737, 21,
17, 25.64598, 21, 18, 30.153956, 21, 19, 23.458814, 22, 1, 17.880614, 22, 2, 26.257996, 22,
3, 21.902978, 22, 4, 29.237483, 22, 5, 27.305361, 22, 6, 25.564741, 22, 7, 27.423738, 22, 8,
27.284202, 22, 9, 26.100016, 22, 10, 24.81979, 22, 11, 26.091215, 22, 12, 29.231088, 22, 13,
21.422142, 22, 14, 27.600349, 22, 15, 24.251346, 22, 16, 26.691911, 22, 17, 28.031678, 22,
18, 25.141837, 22, 19, 24.247653, 23, 1, 3.995247, 23, 2, 27.885453, 23, 3, 25.202678, 23, 4,
27.56778, 23, 5, 20.038175, 23, 6, 27.09378, 23, 7, 28.474479, 23, 8, 24.315505, 23, 9, 24.348003,
23, 10, 26.268426, 23, 11, 28.01939, 23, 12, 24.322967, 23, 13, 29.35389, 23, 14, 25.639698,
23, 15, 26.170224, 23, 16, 26.744998, 23, 17, 24.680595, 23, 18, 25.310422, 23, 19, 29.025513,
23, 20, 25.250504, 24, 1, 24.962648, 24, 2, 28.471088, 24, 3, 28.532157, 24, 4, 27.472327,
24, 5, 20.981399, 24, 6, 24.703986, 24, 7, 27.220383, 24, 8, 26.452425, 24, 9, 27.04022, 24,
10, 21.623458, 24, 11, 26.119236, 24, 12, 25.904002, 24, 13, 27.257536, 24, 14, 29.444097,
24, 15, 26.348127, 24, 16, 30.247531, 24, 17, 26.16532, 24, 18, 23.125314, 24, 19, 26.760823,
25, 1, 17.689883, 25, 2, 27.028643, 25, 3, 25.603946, 25, 4, 24.907982, 25, 5, 24.716491, 25,
6, 26.187502, 25, 7, 28.219994, 25, 8, 23.799895, 25, 9, 26.222162, 25, 10, 27.621528, 25,
11, 23.84554, 25, 12, 27.382108, 25, 13, 28.735976, 25, 14, 24.556881, 25, 15, 26.027312, 25,
16, 25.798956, 25, 17, 30.973714, 25, 18, 25.292561, 25, 19, 24.750077, 26, 1, 4.440027, 26,
2, 21.421521, 26, 3, 26.924377, 26, 4, 26.609901, 26, 5, 31.655153, 26, 6, 30.821074, 26, 7,
27.663298, 26, 8, 29.410261, 26, 9, 27.165355, 26, 10, 26.084233, 26, 11, 28.085346, 26, 12,
26.29459, 26, 13, 28.405811, 26, 14, 27.249535, 26, 15, 26.802775, 26, 16, 23.361783, 26,
17, 26.942586, 26, 18, 26.601919, 26, 19, 26.999081, 27, 1, 10.322134, 27, 2, 24.461698, 27,
3, 27.687287, 27, 4, 26.324498, 27, 5, 27.908453, 27, 6, 23.42245, 27, 7, 27.688202, 27, 8,
24.053484, 27, 9, 26.86212, 27, 10, 25.751357, 27, 11, 27.261373, 27, 12, 25.795417, 27, 13,
28.737066, 27, 14, 28.714362, 27, 15, 25.900367, 27, 16, 25.646309, 27, 17, 28.556094, 27,
18, 23.497909, 27, 19, 28.025475, 28, 1, 8.729696, 28, 2, 28.075273, 28, 3, 27.639193, 28,

```
14, 29.497489, 28, 15, 25.816996, 28, 16, 26.387354, 28, 17, 23.337116, 28, 18, 27.713634,
  28, 19, 27.304305, 29, 1, .987791, 29, 2, 25.244068, 29, 3, 23.846031, 29, 4, 28.222042, 29,
  5, 25.932502, 29, 6, 27.804198, 29, 7, 27.555035, 29, 8, 22.989195, 29, 9, 24.769447, 29, 10,
  23.996075, 29, 11, 25.846155, 29, 12, 26.199277, 29, 13, 28.901275, 29, 14, 28.137472, 29,
  15, 27.957218, 29, 16, 27.225305, 29, 17, 22.801312, 29, 18, 25.172336, 29, 19, 24.781168,
  29, 20, 26.817221, 30, 1, 18.040839, 30, 2, 20.375058, 30, 3, 19.454342, 30, 4, 28.082314, 30,
  5, 28.622506, 30, 6, 34.831803, 30, 7, 25.417457, 30, 8, 23.766261, 30, 9, 27.880763, 30, 10,
  25.313861, 30, 11, 29.098494, 30, 12, 29.160844, 30, 13, 31.923777, 30, 14, 28.109152, 30,
  15, 27.682093, 30, 16, 28.303171, 30, 17, 26.148781, 30, 18, 29.109301, 30, 19, 25.355908};
ss32768Elapsed = {1, 1, .921864, 1, 2, 3.131713, 1, 3, 3.378505, 1, 4, 3.153768, 1, 5, 2.955937, 1,
  6, 2.840469, 1, 7, 3.26345, 1, 8, 2.992568, 1, 9, 3.358107, 1, 10, 2.782871, 1, 11, 3.052726, 1,
  12, 2.967723, 1, 13, 3.64576, 1, 14, 3.162648, 1, 15, 2.932035, 1, 16, 2.992814, 1, 17, 3.280791,
  1, 18, 5.155707, 1, 19, 3.067802, 1, 20, 3.541729, 1, 21, 3.053662, 1, 22, 3.228046, 1, 23,
  3.718416, 1, 24, 3.044173, 1, 25, 6.238628, 1, 26, 3.175418, 1, 27, 4.902081, 1, 28, 2.93183, 1,
  29, 3.846789, 1, 30, 3.236746, 1, 31, 3.114781, 1, 32, 3.105815, 1, 33, 3.616988, 1, 34, 3.466059,
  1, 35, 3.114974, 1, 36, 2.96686, 1, 37, 3.152755, 1, 38, 2.811701, 1, 39, 3.10571, 1, 40, 3.480278,
  1, 41, 3.114812, 1, 42, 3.237811, 1, 43, 3.201793, 1, 44, 8.623924, 1, 45, 3.236455, 1, 46,
  4.321434, 1, 47, 3.480767, 1, 48, 2.882525, 1, 49, 2.699312, 1, 50, 3.435768, 1, 51, 3.322812,
  1, 52, 3.087176, 1, 53, 3.114169, 1, 54, 2.890421, 1, 55, 3.087978, 1, 56, 3.156847, 1, 57,
  3.186025, 1, 58, 3.665967, 1, 59, 2.748892, 1, 60, 3.041805, 1, 61, 3.319754, 1, 62, 4.413714,
  1, 63, 3.070728, 1, 64, 3.204748, 1, 65, 3.053851, 1, 66, 2.891905, 1, 67, 2.957911, 1, 68,
  3.175769, 1, 69, 3.11582, 1, 70, 3.974093, 1, 71, 3.175847, 1, 72, 3.490769, 2, 1, 2.348677, 2,
  2, 3.344748, 2, 3, 3.06331, 2, 4, 3.153049, 2, 5, 4.155415, 2, 6, 3.888736, 2, 7, 2.823437, 2, 8,
  3.210821, 2, 9, 3.009775, 2, 10, 3.179752, 2, 11, 3.387681, 2, 12, 3.412851, 2, 13, 2.938035, 2,
  14, 3.138805, 2, 15, 3.775761, 2, 16, 2.978219, 2, 17, 3.107779, 2, 18, 2.97032, 2, 19, 3.068673,
  2, 20, 3.037702, 2, 21, 3.107836, 2, 22, 3.319378, 2, 23, 2.87501, 2, 24, 2.861411, 2, 25,
  3.009971, 2, 26, 2.865817, 2, 27, 2.965742, 2, 28, 3.036735, 2, 29, 3.0813, 2, 30, 2.894987, 2, 31,
  3.153694, 2, 32, 2.927729, 2, 33, 3.113296, 2, 34, 2.97276, 2, 35, 3.109241, 2, 36, 3.364759, 2,
  37, 3.90481, 2, 38, 2.871118, 2, 39, 3.273778, 2, 40, 3.514277, 2, 41, 3.036888, 2, 42, 3.037107,
  2, 43, 3.082196, 2, 44, 3.614151, 2, 45, 3.585719, 2, 46, 2.965683, 2, 47, 2.925392, 2, 48,
  2.823989, 2, 49, 3.216839, 2, 50, 3.024901, 2, 51, 3.059136, 2, 52, 3.67574, 2, 53, 3.499403, 2,
  54, 3.369838, 2, 55, 3.00973, 2, 56, 3.391842, 2, 57, 3.370162, 2, 58, 3.277846, 2, 59, 3.081806,
  2, 60, 3.225806, 2, 61, 3.061743, 2, 62, 3.946784, 2, 63, 4.305793, 2, 64, 3.036838, 2, 65,
  3.00075, 2, 66, 2.772163, 2, 67, 3.249088, 2, 68, 3.412784, 2, 69, 2.992127, 2, 70, 5.600694, 2,
  71, 4.991714, 2, 72, 4.707578, 3, 1, 5.802639, 3, 2, 3.036168, 3, 3, 3.503782, 3, 4, 3.010368, 3,
  5, 3.038186, 3, 6, 2.92647, 3, 7, 3.036137, 3, 8, 2.793201, 3, 9, 2.86041, 3, 10, 3.533749, 3, 11,
  3.951837, 3, 12, 3.29779, 3, 13, 3.009749, 3, 14, 3.729924, 3, 15, 3.178858, 3, 16, 3.107792, 3,
  17, 3.249767, 3, 18, 3.081927, 3, 19, 3.730163, 3, 20, 3.036803, 3, 21, 2.823254, 3, 22, 3.416503,
  3, 23, 3.107735, 3, 24, 3.576375, 3, 25, 3.292797, 3, 26, 3.391784, 3, 27, 3.036247, 3, 28,
  3.108488, 3, 29, 2.765876, 3, 30, 2.91973, 3, 31, 4.331342, 3, 32, 2.965315, 3, 33, 3.107777, 3,
  34, 4.084705, 3, 35, 2.9658, 3, 36, 3.782816, 3, 37, 3.107811, 3, 38, 2.939443, 3, 39, 3.296779,
  3, 40, 3.431785, 3, 41, 3.614129, 3, 42, 3.320107, 3, 43, 2.96525, 3, 44, 3.282758, 3, 45,
  4.257277, 3, 46, 2.965761, 3, 47, 3.391112, 3, 48, 3.077568, 3, 49, 3.39277, 3, 50, 2.965237, 3,
  51, 4.089654, 3, 52, 3.370091, 3, 53, 2.907022, 3, 54, 3.11085, 3, 55, 3.294613, 3, 56, 2.937764,
  3, 57, 3.729914, 3, 58, 3.036497, 3, 59, 3.730068, 3, 60, 3.112797, 3, 61, 3.679708, 3, 62,
  6.340693, 3, 63, 3.586107, 3, 64, 2.973308, 3, 65, 6.151023, 3, 66, 3.817061, 3, 67, 5.196741, 3,
  68, 3.154107, 3, 69, 2.99079, 3, 70, 3.096277, 3, 71, 3.256841, 4, 1, 3.442434, 4, 2, 2.96575, 4,
  3, 3.107717, 4, 4, 3.623091, 4, 5, 3.083761, 4, 6, 2.965154, 4, 7, 3.190814, 4, 8, 3.009699, 4,
  9, 3.579194, 4, 10, 3.082077, 4, 11, 3.118736, 4, 12, 2.937621, 4, 13, 2.96575, 4, 14, 2.915011,
  4, 15, 2.909961, 4, 16, 3.03697, 4, 17, 3.249846, 4, 18, 3.604779, 4, 19, 2.895123, 4, 20,
  4.101754, 4, 21, 3.392056, 4, 22, 3.249817, 4, 23, 2.914196, 4, 24, 5.843741, 4, 25, 3.153715,
  4, 26, 2.825534, 4, 27, 3.179264, 4, 28, 3.676076, 4, 29, 2.851792, 4, 30, 3.154169, 4, 31,
  3.224974, 4, 32, 3.249114, 4, 33, 2.894072, 4, 34, 3.084694, 4, 35, 2.690829, 4, 36, 2.965796,
  4, 37, 3.107818, 4, 38, 3.462779, 4, 39, 3.510708, 4, 40, 3.009729, 4, 41, 3.154096, 4, 42,
  2.999813, 4, 43, 3.375778, 4, 44, 3.063248, 4, 45, 3.176689, 4, 46, 3.320762, 4, 47, 4.203667,
  4, 48, 2.996164, 4, 49, 3.657723, 4, 50, 4.103028, 4, 51, 2.753016, 4, 52, 3.178789, 4, 53,
  3.370068, 4, 54, 3.249825, 4, 55, 3.131126, 4, 56, 3.451933, 4, 57, 2.965241, 4, 58, 3.320813,
  4, 59, 3.153737, 4, 60, 3.193636, 4, 61, 3.207741, 4, 62, 3.107786, 4, 63, 3.144936, 4, 64,
  3.434762, 4, 65, 6.79036, 4, 66, 3.122898, 4, 67, 4.08976, 4, 68, 3.297755, 4, 69, 2.966238, 4,
  70, 3.364756, 4, 71, 3.107819, 4, 72, 4.305454, 5, 1, .286391, 5, 2, 3.585744, 5, 3, 3.369771, 5,
  4, 3.009876, 5, 5, 2.866146, 5, 6, 3.51375, 5, 7, 2.721975, 5, 8, 3.279768, 5, 9, 4.479495, 5, 10,
  3.036173, 5, 11, 3.189788, 5, 12, 3.225765, 5, 13, 4.669778, 5, 14, 5.806679, 5, 15, 3.585788,
  5, 16, 3.010039, 5, 17, 3.320782, 5, 18, 3.321508, 5, 19, 4.313984, 5, 20, 3.153378, 5, 21,
  3.122805, 5, 22, 2.997758, 5, 23, 2.974075, 5, 24, 3.399783, 5, 25, 3.178787, 5, 26, 3.694075,
  5, 27, 3.009757, 5, 28, 3.829705, 5, 29, 3.490716, 5, 30, 3.050765, 5, 31, 3.392867, 5, 32,
  2.9969, 5, 33, 3.120723, 5, 34, 3.087767, 5, 35, 3.2507, 5, 36, 3.062797, 5, 37, 3.579463, 5, 38,
  3.88787, 5, 39, 3.006074, 5, 40, 3.301729, 5, 41, 3.153784, 5, 42, 3.07586, 5, 43, 2.859362, 5,
  44, 2.916774, 5, 45, 3.08374, 5, 46, 3.80578, 5, 47, 4.847703, 5, 48, 2.824136, 5, 49, 3.874155,
  5, 50, 4.115135, 5, 51, 2.933823, 5, 52, 3.205756, 5, 53, 3.107753, 5, 54, 3.192984, 5, 55,
  3.371555, 5, 56, 3.062095, 5, 57, 3.225756, 5, 58, 3.107894, 5, 59, 2.902349, 5, 60, 3.298494, 5,
  61, 3.107707, 5, 62, 2.965241, 5, 63, 3.679746, 5, 64, 3.10781, 5, 65, 4.03972, 5, 66, 3.146686,
  5, 67, 3.118837, 5, 68, 3.249819, 5, 69, 3.729766, 5, 70, 3.108115, 5, 71, 2.966205, 5, 72,
  3.221757, 6, 1, .074442, 6, 2, 3.036047, 6, 3, 3.320744, 6, 4, 3.08174, 6, 5, 5.129815, 6, 6,
```

6, 18, 2.976171, 6, 19, 3.442397, 6, 20, 3.178125, 6, 21, 3.537827, 6, 22, 2.933607, 6, 23,
3.010468, 6, 24, 2.83834, 6, 25, 3.80174, 6, 26, 2.985083, 6, 27, 3.177801, 6, 28, 3.51429, 6,
29, 3.145352, 6, 30, 3.15378, 6, 31, 3.0824, 6, 32, 3.372794, 6, 33, 2.904806, 6, 34, 2.996622,
6, 35, 2.965795, 6, 36, 3.210713, 6, 37, 4.812393, 6, 38, 4.252748, 6, 39, 3.222519, 6, 40,
3.178781, 6, 41, 3.040404, 6, 42, 3.066768, 6, 43, 3.179803, 6, 44, 3.056941, 6, 45, 3.60481, 6,
46, 3.40282, 6, 47, 2.894814, 6, 48, 3.131227, 6, 49, 3.178837, 6, 50, 3.13251, 6, 51, 2.978773,
6, 52, 2.86592, 6, 53, 3.29812, 6, 54, 3.178696, 6, 55, 3.045718, 6, 56, 2.996239, 6, 57,
3.297798, 6, 58, 3.723297, 6, 59, 3.226731, 6, 60, 3.10171, 6, 61, 4.016038, 6, 62, 3.452099,
6, 63, 3.252801, 6, 64, 2.981862, 6, 65, 3.107892, 6, 66, 3.162571, 6, 67, 3.074013, 6, 68,
3.633331, 6, 69, 6.257991, 6, 70, 3.876039, 6, 71, 3.081701, 6, 72, 4.574117, 7, 1, .253367, 7,
2, 3.442533, 7, 3, 3.102031, 7, 4, 3.037751, 7, 5, 2.839453, 7, 6, 3.373041, 7, 7, 2.974219, 7,
8, 3.534745, 7, 9, 3.38698, 7, 10, 3.036765, 7, 11, 3.178762, 7, 12, 3.105794, 7, 13, 3.476731,
7, 14, 3.04569, 7, 15, 2.904272, 7, 16, 3.467753, 7, 17, 3.656849, 7, 18, 4.493699, 7, 19,
3.12572, 7, 20, 3.049713, 7, 21, 3.011625, 7, 22, 3.441752, 7, 23, 3.6537, 7, 24, 3.124725,
7, 25, 3.480971, 7, 26, 3.107075, 7, 27, 4.956199, 7, 28, 2.633314, 7, 29, 4.323757, 7, 30,
3.346682, 7, 31, 3.129725, 7, 32, 3.152709, 7, 33, 3.182758, 7, 34, 3.571755, 7, 35, 3.036896,
7, 36, 3.077696, 7, 37, 3.243785, 7, 38, 3.370053, 7, 39, 2.704285, 7, 40, 4.516426, 7, 41,
2.965852, 7, 42, 3.246702, 7, 43, 3.215301, 7, 44, 3.158753, 7, 45, 2.966166, 7, 46, 3.180672,
7, 47, 3.380669, 7, 48, 2.965676, 7, 49, 3.999826, 7, 50, 3.110881, 7, 51, 3.082029, 7, 52,
3.081671, 7, 53, 3.158777, 7, 54, 3.90669, 7, 55, 2.865774, 7, 56, 3.078696, 7, 57, 3.746681, 7,
58, 3.500058, 7, 59, 3.09772, 7, 60, 3.886811, 7, 61, 3.324995, 7, 62, 3.791666, 7, 63, 4.04264,
7, 64, 3.987783, 7, 65, 3.127745, 7, 66, 3.139711, 7, 67, 2.785495, 7, 68, 2.794214, 7, 69,
3.138785, 7, 70, 3.081706, 7, 71, 4.196988, 7, 72, 3.251118, 8, 1, 3.639093, 8, 2, 3.036015, 8,
3, 3.041264, 8, 4, 3.746669, 8, 5, 2.997758, 8, 6, 3.070725, 8, 7, 4.974482, 8, 8, 4.098395, 8,
9, 4.089081, 8, 10, 3.228595, 8, 11, 3.510724, 8, 12, 3.52072, 8, 13, 3.447343, 8, 14, 4.071964,
8, 15, 3.070473, 8, 16, 3.539534, 8, 17, 2.829627, 8, 18, 3.094731, 8, 19, 3.240786, 8, 20,
3.194456, 8, 21, 3.092241, 8, 22, 2.824613, 8, 23, 3.53345, 8, 24, 3.316754, 8, 25, 2.969859,
8, 26, 3.176812, 8, 27, 3.082084, 8, 28, 3.107654, 8, 29, 2.922315, 8, 30, 3.474062, 8, 31,
3.614771, 8, 32, 3.043131, 8, 33, 3.913847, 8, 34, 2.828963, 8, 35, 3.10769, 8, 36, 2.845844,
8, 37, 3.483707, 8, 38, 2.966084, 8, 39, 4.16189, 8, 40, 3.370822, 8, 41, 3.159778, 8, 42,
3.310155, 8, 43, 3.005294, 8, 44, 3.188931, 8, 45, 3.178843, 8, 46, 2.894426, 8, 47, 3.917662,
8, 48, 4.228654, 8, 49, 2.965967, 8, 50, 3.164711, 8, 51, 3.20105, 8, 52, 3.290689, 8, 53,
3.254744, 8, 54, 3.544686, 8, 55, 2.945733, 8, 56, 3.225796, 8, 57, 3.262881, 8, 58, 3.256718,
8, 59, 3.393547, 8, 60, 2.695112, 8, 61, 2.894092, 8, 62, 3.082765, 8, 63, 3.330725, 8, 64,
3.184679, 8, 65, 3.467739, 8, 66, 3.287617, 8, 67, 3.744439, 8, 68, 2.865704, 8, 69, 3.24567, 8,
70, 3.052022, 8, 71, 3.022104, 8, 72, 3.075721, 9, 1, 3.028695, 9, 2, 3.110837, 9, 3, 2.903839,
9, 4, 3.364968, 9, 5, 3.391786, 9, 6, 2.937849, 9, 7, 3.098646, 9, 8, 3.810583, 9, 9, 3.388857,
9, 10, 2.999737, 9, 11, 4.131085, 9, 12, 3.213864, 9, 13, 2.991082, 9, 14, 3.037042, 9, 15,
2.966208, 9, 16, 3.213931, 9, 17, 2.937148, 9, 18, 3.293286, 9, 19, 3.009731, 9, 20, 3.362767,
9, 21, 3.081765, 9, 22, 3.526666, 9, 23, 3.196542, 9, 24, 3.107488, 9, 25, 3.297804, 9, 26,
3.467731, 9, 27, 3.292722, 9, 28, 3.108735, 9, 29, 3.883728, 9, 30, 3.076716, 9, 31, 3.052645,
9, 32, 5.069209, 9, 33, 5.304621, 9, 34, 4.027694, 9, 35, 3.010089, 9, 36, 3.493202, 9, 37,
3.146056, 9, 38, 3.229781, 9, 39, 3.80105, 9, 40, 2.777229, 9, 41, 3.055815, 9, 42, 3.149759,
9, 43, 2.965127, 9, 44, 4.918065, 9, 45, 4.652757, 9, 46, 5.061295, 9, 47, 3.136763, 9, 48,
3.722637, 9, 49, 3.754682, 9, 50, 3.063743, 9, 51, 2.972479, 9, 52, 3.036933, 9, 53, 4.640786,
9, 54, 2.981081, 9, 55, 2.965681, 9, 56, 2.92692, 9, 57, 3.616723, 9, 58, 3.225083, 9, 59,
4.04999, 9, 60, 5.283852, 9, 61, 3.081763, 9, 62, 3.670664, 9, 63, 3.460869, 9, 64, 3.225774,
9, 65, 3.532892, 9, 66, 3.520536, 9, 67, 4.473809, 9, 68, 3.423145, 9, 69, 3.988717, 9, 70,
4.377603, 10, 1, 4.872899, 10, 2, 3.181081, 10, 3, 3.178023, 10, 4, 3.735799, 10, 5, 3.861229,
10, 6, 3.042659, 10, 7, 2.941367, 10, 8, 3.064815, 10, 9, 2.836016, 10, 10, 2.865523, 10, 11,
3.284713, 10, 12, 2.865294, 10, 13, 3.006978, 10, 14, 4.238677, 10, 15, 3.326753, 10, 16, 3.03063,
10, 17, 3.230835, 10, 18, 3.243379, 10, 19, 3.678259, 10, 20, 2.894717, 10, 21, 3.624641, 10,
22, 2.895462, 10, 23, 2.774572, 10, 24, 2.925896, 10, 25, 3.260439, 10, 26, 2.938372, 10, 27,
3.044679, 10, 28, 3.918288, 10, 29, 3.153871, 10, 30, 3.128703, 10, 31, 2.965023, 10, 32,
3.392932, 10, 33, 3.036843, 10, 34, 2.824122, 10, 35, 4.046695, 10, 36, 3.028718, 10, 37,
3.107971, 10, 38, 2.988659, 10, 39, 3.636025, 10, 40, 3.658748, 10, 41, 3.230854, 10, 42,
3.212765, 10, 43, 3.110439, 10, 44, 2.611326, 10, 45, 3.739962, 10, 46, 2.986235, 10, 47,
3.224787, 10, 48, 3.089798, 10, 49, 3.036736, 10, 50, 3.039871, 10, 51, 3.095819, 10, 52,
3.302937, 10, 53, 3.210462, 10, 54, 3.605087, 10, 55, 3.299057, 10, 56, 2.96568, 10, 57, 5.121522,
10, 58, 3.137828, 10, 59, 2.96598, 10, 60, 3.036799, 10, 61, 3.258687, 10, 62, 3.178263, 10,
63, 3.401786, 10, 64, 5.058924, 10, 65, 3.307335, 10, 66, 3.107033, 10, 67, 3.494036, 10,
68, 3.044564, 10, 69, 4.764259, 10, 70, 4.480456, 10, 71, 3.808042, 11, 1, 2.910027, 11, 2,
3.215191, 11, 3, 3.051985, 11, 4, 3.168598, 11, 5, 3.196007, 11, 6, 2.867802, 11, 7, 3.127374,
11, 8, 3.13477, 11, 9, 3.555723, 11, 10, 3.125741, 11, 11, 3.759965, 11, 12, 3.042419, 11,
13, 3.212863, 11, 14, 2.970434, 11, 15, 2.794313, 11, 16, 2.84223, 11, 17, 3.126106, 11, 18,
3.276476, 11, 19, 3.02592, 11, 20, 2.936918, 11, 21, 3.395903, 11, 22, 2.894806, 11, 23, 3.695805,
11, 24, 3.394743, 11, 25, 3.044046, 11, 26, 3.935479, 11, 27, 3.096906, 11, 28, 3.123088, 11,
29, 5.082517, 11, 30, 3.230639, 11, 31, 3.233566, 11, 32, 3.443158, 11, 33, 4.277979, 11, 34,
2.85482, 11, 35, 3.117089, 11, 36, 4.039082, 11, 37, 2.965366, 11, 38, 3.243727, 11, 39, 3.103763,
11, 40, 3.107736, 11, 41, 4.009421, 11, 42, 3.178188, 11, 43, 4.19198, 11, 44, 3.422858, 11,
45, 5.481692, 11, 46, 2.976043, 11, 47, 3.792966, 11, 48, 3.757541, 11, 49, 2.681865, 11,
50, 3.506741, 11, 51, 3.083747, 11, 52, 3.651789, 11, 53, 2.624363, 11, 54, 3.221252, 11,
55, 4.052104, 11, 56, 3.20675, 11, 57, 3.298427, 11, 58, 3.377049, 11, 59, 2.758419, 11, 60,

```
11, 71, 2.965337, 12, 1, 3.010062, 12, 2, 3.481722, 12, 3, 3.013099, 12, 4, 3.153758, 12, 5,
3.39882, 12, 6, 4.449641, 12, 7, 2.86621, 12, 8, 4.300596, 12, 9, 3.009735, 12, 10, 3.151637,
12, 11, 3.079038, 12, 12, 3.15194, 12, 13, 3.591716, 12, 14, 3.638698, 12, 15, 2.859786, 12,
16, 3.591007, 12, 17, 3.948457, 12, 18, 3.081695, 12, 19, 2.935209, 12, 20, 2.983783, 12, 21,
3.235972, 12, 22, 3.080682, 12, 23, 3.744629, 12, 24, 4.910126, 12, 25, 3.209931, 12, 26,
4.316243, 12, 27, 3.069795, 12, 28, 2.937083, 12, 29, 3.135027, 12, 30, 3.001064, 12, 31,
3.948056, 12, 32, 5.327896, 12, 33, 3.087777, 12, 34, 3.339741, 12, 35, 2.847757, 12, 36,
3.240787, 12, 37, 3.010336, 12, 38, 3.298046, 12, 39, 3.205706, 12, 40, 3.691727, 12, 41,
2.988427, 12, 42, 3.277716, 12, 43, 3.036569, 12, 44, 3.335683, 12, 45, 3.628601, 12, 46,
3.077316, 12, 47, 3.589733, 12, 48, 3.959695, 12, 49, 3.036093, 12, 50, 2.90447, 12, 51, 2.965768,
12, 52, 3.190763, 12, 53, 3.14615, 12, 54, 3.172941, 12, 55, 3.393987, 12, 56, 3.796995, 12,
57, 3.657377, 12, 58, 3.04803, 12, 59, 3.430639, 12, 60, 3.107369, 12, 61, 3.005084, 12, 62,
2.936996, 12, 63, 3.62982, 12, 64, 3.153767, 12, 65, 3.532733, 12, 66, 2.977229, 12, 67, 3.071555,
12, 68, 2.897651, 12, 69, 3.228708, 12, 70, 3.226036, 12, 71, 3.48324, 13, 1, .058288, 13, 2,
2.982904, 13, 3, 4.386992, 13, 4, 2.909866, 13, 5, 3.205236, 13, 6, 3.082051, 13, 7, 3.677653,
13, 8, 3.037173, 13, 9, 3.220755, 13, 10, 2.977272, 13, 11, 2.979807, 13, 12, 3.060793, 13,
13, 3.318019, 13, 14, 3.538821, 13, 15, 3.289683, 13, 16, 3.116207, 13, 17, 3.063106, 13,
18, 3.069368, 13, 19, 3.203339, 13, 20, 4.393734, 13, 21, 3.158876, 13, 22, 2.700497, 13,
23, 3.98474, 13, 24, 4.730684, 13, 25, 2.973105, 13, 26, 2.610889, 13, 27, 2.989559, 13, 28,
3.165715, 13, 29, 3.26753, 13, 30, 2.973384, 13, 31, 2.968234, 13, 32, 2.965208, 13, 33, 4.399028,
13, 34, 4.130533, 13, 35, 3.741733, 13, 36, 3.237916, 13, 37, 3.715035, 13, 38, 3.107102,
13, 39, 4.235796, 13, 40, 3.03663, 13, 41, 4.614847, 13, 42, 3.916346, 13, 43, 3.840706, 13,
44, 3.987718, 13, 45, 3.142762, 13, 46, 3.178778, 13, 47, 3.042954, 13, 48, 3.918895, 13,
49, 3.444679, 13, 50, 3.474352, 13, 51, 2.897413, 13, 52, 3.289721, 13, 53, 4.168831, 13,
54, 3.846665, 13, 55, 3.062749, 13, 56, 3.384756, 13, 57, 3.116672, 13, 58, 3.162001, 13,
59, 2.894993, 13, 60, 3.070052, 13, 61, 3.392029, 13, 62, 3.175982, 13, 63, 5.073662, 13,
64, 3.113809, 13, 65, 3.038831, 13, 66, 3.178733, 13, 67, 3.165693, 13, 68, 2.986125, 13,
69, 3.085735, 13, 70, 5.133665, 13, 71, 2.993675, 14, 1, 3.140016, 14, 2, 2.984284, 14, 3,
3.194098, 14, 4, 3.521849, 14, 5, 3.323896, 14, 6, 8.072225, 14, 7, 3.195755, 14, 8, 3.061671,
14, 9, 2.990766, 14, 10, 3.138696, 14, 11, 3.065785, 14, 12, 2.704455, 14, 13, 2.863196, 14,
14, 3.23687, 14, 15, 3.230668, 14, 16, 3.129419, 14, 17, 3.072075, 14, 18, 3.136084, 14, 19,
3.729659, 14, 20, 2.994023, 14, 21, 5.867479, 14, 22, 3.010134, 14, 23, 2.910382, 14, 24,
3.530637, 14, 25, 3.034778, 14, 26, 3.419717, 14, 27, 3.330894, 14, 28, 3.366769, 14, 29,
3.268664, 14, 30, 3.081798, 14, 31, 3.357652, 14, 32, 3.107015, 14, 33, 2.937711, 14, 34,
3.403778, 14, 35, 3.271748, 14, 36, 3.19475, 14, 37, 3.514078, 14, 38, 4.834803, 14, 39, 3.122384,
14, 40, 3.228086, 14, 41, 3.178149, 14, 42, 2.983092, 14, 43, 3.06316, 14, 44, 2.985084, 14,
45, 3.110764, 14, 46, 2.849898, 14, 47, 3.682563, 14, 48, 3.226372, 14, 49, 3.339115, 14,
50, 3.435605, 14, 51, 2.937976, 14, 52, 3.151703, 14, 53, 3.523709, 14, 54, 2.864437, 14,
55, 3.729709, 14, 56, 3.379058, 14, 57, 3.209676, 14, 58, 3.444734, 14, 59, 3.907705, 14,
60, 2.681949, 14, 61, 3.401688, 14, 62, 3.686812, 14, 63, 3.077694, 14, 64, 3.107702, 14,
65, 3.121079, 14, 66, 2.858951, 14, 67, 3.060309, 14, 68, 2.894357, 14, 69, 3.648139, 14,
70, 8.397963, 14, 71, 3.285808, 15, 1, 4.172294, 15, 2, 3.079733, 15, 3, 3.181681, 15, 4,
2.897459, 15, 5, 3.56684, 15, 6, 2.928417, 15, 7, 3.225826, 15, 8, 3.153802, 15, 9, 2.974194,
15, 10, 3.586915, 15, 11, 3.009823, 15, 12, 3.010161, 15, 13, 4.065918, 15, 14, 3.107707,
15, 15, 3.475715, 15, 16, 3.243199, 15, 17, 2.944733, 15, 18, 3.510037, 15, 19, 3.021822,
15, 20, 4.023716, 15, 21, 3.057806, 15, 22, 4.63286, 15, 23, 3.939697, 15, 24, 4.505006, 15,
25, 3.420106, 15, 26, 3.048704, 15, 27, 2.913212, 15, 28, 3.749027, 15, 29, 3.357118, 15,
30, 3.029392, 15, 31, 3.224605, 15, 32, 3.153099, 15, 33, 3.289734, 15, 34, 4.990876, 15,
35, 5.060192, 15, 36, 3.316801, 15, 37, 3.128665, 15, 38, 3.108925, 15, 39, 3.297716, 15,
40, 3.001708, 15, 41, 2.77773, 15, 42, 3.127348, 15, 43, 2.917088, 15, 44, 3.064818, 15, 45,
3.546791, 15, 46, 2.910126, 15, 47, 4.23005, 15, 48, 3.860684, 15, 49, 3.343904, 15, 50,
3.480698, 15, 51, 2.900195, 15, 52, 3.118044, 15, 53, 3.055825, 15, 54, 2.895017, 15, 55,
3.955708, 15, 56, 4.672701, 15, 57, 5.209742, 15, 58, 4.470009, 15, 59, 2.753112, 15, 60,
3.084544, 15, 61, 2.859303, 15, 62, 3.062692, 15, 63, 3.118368, 15, 64, 3.153701, 15, 65,
3.123716, 15, 66, 3.21492, 15, 67, 2.965739, 15, 68, 3.930507, 15, 69, 5.654, 15, 70, 3.090734,
16, 1, 3.007464, 16, 2, 2.894842, 16, 3, 3.196943, 16, 4, 3.702916, 16, 5, 3.036146, 16, 6,
3.578737, 16, 7, 3.597479, 16, 8, 3.073693, 16, 9, 2.966042, 16, 10, 2.894809, 16, 11, 2.978764,
16, 12, 2.610213, 16, 13, 3.902082, 16, 14, 4.055743, 16, 15, 3.114061, 16, 16, 3.125759, 16,
17, 3.2538, 16, 18, 5.202641, 16, 19, 3.712722, 16, 20, 3.215716, 16, 21, 3.81089, 16, 22,
3.570135, 16, 23, 2.941532, 16, 24, 3.061389, 16, 25, 3.010399, 16, 26, 4.101663, 16, 27,
4.944754, 16, 28, 2.955768, 16, 29, 3.043602, 16, 30, 3.107136, 16, 31, 2.652672, 16, 32,
2.965415, 16, 33, 2.907063, 16, 34, 2.84828, 16, 35, 3.189402, 16, 36, 4.03968, 16, 37, 3.062073,
16, 38, 3.036702, 16, 39, 4.719048, 16, 40, 2.864955, 16, 41, 3.178665, 16, 42, 2.966805,
16, 43, 2.998802, 16, 44, 2.973793, 16, 45, 3.225838, 16, 46, 3.541689, 16, 47, 2.899399,
16, 48, 2.973067, 16, 49, 2.995061, 16, 50, 3.076741, 16, 51, 3.05378, 16, 52, 3.342863, 16,
53, 3.137113, 16, 54, 2.998757, 16, 55, 2.84618, 16, 56, 3.541767, 16, 57, 5.340636, 16,
58, 2.830497, 16, 59, 4.970896, 16, 60, 3.922712, 16, 61, 3.498562, 16, 62, 2.994692, 16,
63, 2.98875, 16, 64, 3.001662, 16, 65, 3.476919, 16, 66, 3.044993, 16, 67, 3.127699, 16,
68, 3.317723, 16, 69, 3.214706, 16, 70, 2.905215, 16, 71, 4.931382, 17, 1, 3.200719, 17, 2,
3.360734, 17, 3, 3.116871, 17, 4, 3.047748, 17, 5, 3.608734, 17, 6, 2.894361, 17, 7, 3.459399,
17, 8, 3.04237, 17, 9, 3.12772, 17, 10, 3.391041, 17, 11, 3.519171, 17, 12, 3.188853, 17, 13,
3.107894, 17, 14, 4.195584, 17, 15, 3.470718, 17, 16, 4.471746, 17, 17, 5.795633, 17, 18,
3.15606, 17, 19, 3.179732, 17, 20, 2.886126, 17, 21, 3.28063, 17, 22, 2.903969, 17, 23, 3.159096,
```

17, 34, 3.497594, 17, 35, 4.599804, 17, 36, 3.271094, 17, 37, 3.121914, 17, 38, 3.079696,
17, 39, 3.947627, 17, 40, 3.107784, 17, 41, 3.5206, 17, 42, 3.038433, 17, 43, 3.815921, 17,
44, 3.589095, 17, 45, 3.199759, 17, 46, 3.066131, 17, 47, 2.904936, 17, 48, 4.784607, 17,
49, 5.796641, 17, 50, 4.717116, 17, 51, 3.010135, 17, 52, 2.990369, 17, 53, 3.297712, 17,
54, 2.965877, 17, 55, 3.522984, 17, 56, 3.172887, 17, 57, 2.9381, 17, 58, 3.044726, 17, 59,
3.249758, 17, 60, 3.7191, 17, 61, 3.180847, 17, 62, 3.818686, 17, 63, 3.334735, 17, 64, 2.965034,
17, 65, 3.297682, 17, 66, 3.223667, 17, 67, 3.249895, 17, 68, 2.76485, 17, 69, 3.178505, 17,
70, 2.993347, 17, 71, 2.690781, 18, 1, 3.498198, 18, 2, 2.984864, 18, 3, 3.469641, 18, 4,
2.967675, 18, 5, 2.85873, 18, 6, 2.965168, 18, 7, 3.799798, 18, 8, 3.469096, 18, 9, 3.110343,
18, 10, 4.154027, 18, 11, 3.976777, 18, 12, 3.220716, 18, 13, 2.866536, 18, 14, 3.199644,
18, 15, 3.067654, 18, 16, 3.809783, 18, 17, 3.049074, 18, 18, 3.63743, 18, 19, 3.66708, 18,
20, 3.449526, 18, 21, 3.512687, 18, 22, 3.759607, 18, 23, 4.305983, 18, 24, 2.999799, 18,
25, 4.612432, 18, 26, 4.472863, 18, 27, 3.240756, 18, 28, 2.968438, 18, 29, 3.238644, 18,
30, 2.969077, 18, 31, 3.796584, 18, 32, 2.96561, 18, 33, 4.052939, 18, 34, 5.055358, 18, 35,
4.286309, 18, 36, 3.366772, 18, 37, 2.965753, 18, 38, 3.257687, 18, 39, 3.309719, 18, 40,
4.588699, 18, 41, 3.462928, 18, 42, 3.37658, 18, 43, 3.054831, 18, 44, 5.285704, 18, 45,
3.149662, 18, 46, 3.06465, 18, 47, 3.036789, 18, 48, 3.009799, 18, 49, 3.487778, 18, 50,
5.018672, 18, 51, 2.893717, 18, 52, 3.13229, 18, 53, 3.626582, 18, 54, 4.111014, 18, 55,
2.900123, 18, 56, 2.915618, 18, 57, 2.96586, 18, 58, 3.46375, 18, 59, 3.850672, 18, 60, 3.87175,
18, 61, 3.469005, 18, 62, 3.857089, 18, 63, 3.503683, 18, 64, 3.220871, 18, 65, 3.381366,
18, 66, 3.114149, 18, 67, 3.729884, 18, 68, 2.774505, 18, 69, 3.077544, 18, 70, 3.523724,
19, 1, 3.081307, 19, 2, 3.320184, 19, 3, 3.141707, 19, 4, 3.640457, 19, 5, 3.622608, 19, 6,
2.919769, 19, 7, 3.72476, 19, 8, 2.965953, 19, 9, 3.035972, 19, 10, 3.010154, 19, 11, 3.273095,
19, 12, 2.977652, 19, 13, 3.477735, 19, 14, 4.197705, 19, 15, 2.64963, 19, 16, 3.992481,
19, 17, 3.112096, 19, 18, 4.150877, 19, 19, 2.847733, 19, 20, 3.824717, 19, 21, 3.593927,
19, 22, 2.879185, 19, 23, 3.152671, 19, 24, 3.058592, 19, 25, 3.357659, 19, 26, 2.988891,
19, 27, 2.838101, 19, 28, 3.063699, 19, 29, 3.153029, 19, 30, 3.107721, 19, 31, 2.96788,
19, 32, 3.516746, 19, 33, 3.121684, 19, 34, 3.826756, 19, 35, 2.896453, 19, 36, 3.333078,
19, 37, 2.965081, 19, 38, 2.940304, 19, 39, 4.244105, 19, 40, 2.957764, 19, 41, 4.766541,
19, 42, 2.916704, 19, 43, 2.823922, 19, 44, 3.184789, 19, 45, 3.036706, 19, 46, 2.838725,
19, 47, 3.435665, 19, 48, 5.364692, 19, 49, 3.154032, 19, 50, 3.082082, 19, 51, 3.419699,
19, 52, 2.895012, 19, 53, 4.78808, 19, 54, 5.298267, 19, 55, 2.885755, 19, 56, 3.223893,
19, 57, 2.982536, 19, 58, 4.22488, 19, 59, 4.243783, 19, 60, 3.037249, 19, 61, 2.632404,
19, 62, 4.061924, 19, 63, 3.12387, 19, 64, 2.937719, 19, 65, 3.047642, 19, 66, 3.49781, 19,
67, 3.129701, 19, 68, 4.152685, 19, 69, 3.074333, 19, 70, 5.632948, 20, 1, 3.107751, 20, 2,
3.099737, 20, 3, 2.824029, 20, 4, 2.965798, 20, 5, 4.319379, 20, 6, 2.793566, 20, 7, 2.935727,
20, 8, 2.778767, 20, 9, 3.771644, 20, 10, 4.228771, 20, 11, 3.028302, 20, 12, 3.03659, 20,
13, 3.243747, 20, 14, 4.002067, 20, 15, 3.126709, 20, 16, 4.974672, 20, 17, 3.794522, 20,
18, 3.107742, 20, 19, 2.614373, 20, 20, 3.872069, 20, 21, 3.60468, 20, 22, 3.908724, 20, 23,
4.154464, 20, 24, 2.713782, 20, 25, 2.965689, 20, 26, 3.781946, 20, 27, 3.107816, 20, 28,
4.808703, 20, 29, 3.218744, 20, 30, 2.940139, 20, 31, 3.501685, 20, 32, 3.036761, 20, 33,
3.746867, 20, 34, 3.213696, 20, 35, 3.406038, 20, 36, 3.074933, 20, 37, 3.872765, 20, 38,
3.015591, 20, 39, 4.010545, 20, 40, 3.653031, 20, 41, 3.109703, 20, 42, 3.295758, 20, 43,
3.183061, 20, 44, 2.994979, 20, 45, 4.650172, 20, 46, 2.895158, 20, 47, 3.461755, 20, 48,
3.036716, 20, 49, 3.522227, 20, 50, 2.96581, 20, 51, 4.727602, 20, 52, 2.77593, 20, 53, 3.693695,
20, 54, 3.118758, 20, 55, 2.823781, 20, 56, 3.06703, 20, 57, 3.284006, 20, 58, 3.107894, 20,
59, 3.296689, 20, 60, 3.142944, 20, 61, 5.236704, 20, 62, 3.497658, 20, 63, 3.044674, 20,
64, 4.316664, 20, 65, 3.476635, 20, 66, 3.429664, 20, 67, 3.404698, 20, 68, 3.515723, 20,
69, 3.108063, 20, 70, 3.816624, 21, 1, 3.081676, 21, 2, 3.908554, 21, 3, 3.410978, 21, 4,
3.474742, 21, 5, 3.599067, 21, 6, 3.533825, 21, 7, 3.344362, 21, 8, 3.108241, 21, 9, 3.683805,
21, 10, 3.855937, 21, 11, 3.525398, 21, 12, 3.191126, 21, 13, 2.897513, 21, 14, 3.128824,
21, 15, 4.42705, 21, 16, 3.573777, 21, 17, 3.201704, 21, 18, 3.258563, 21, 19, 2.938318, 21,
20, 3.272766, 21, 21, 3.320638, 21, 22, 3.92372, 21, 23, 3.082102, 21, 24, 3.159277, 21, 25,
3.402911, 21, 26, 3.470705, 21, 27, 3.081254, 21, 28, 3.079686, 21, 29, 3.083812, 21, 30,
3.010355, 21, 31, 3.05301, 21, 32, 3.17853, 21, 33, 3.390742, 21, 34, 3.746714, 21, 35, 3.226074,
21, 36, 4.887674, 21, 37, 2.613621, 21, 38, 2.930146, 21, 39, 3.269675, 21, 40, 3.462778,
21, 41, 2.909277, 21, 42, 3.126799, 21, 43, 4.322117, 21, 44, 2.930869, 21, 45, 3.064786,
21, 46, 3.601766, 21, 47, 3.089713, 21, 48, 3.205391, 21, 49, 3.232776, 21, 50, 3.817953,
21, 51, 3.156067, 21, 52, 3.707874, 21, 53, 3.925897, 21, 54, 3.769335, 21, 55, 3.08212, 21,
56, 3.383385, 21, 57, 3.890174, 21, 58, 3.32291, 21, 59, 3.655717, 21, 60, 4.104744, 21, 61,
3.045721, 21, 62, 4.070622, 21, 63, 3.402402, 21, 64, 3.009745, 21, 65, 3.728236, 21, 66,
3.322645, 21, 67, 3.153698, 21, 68, 2.937841, 21, 69, 3.144304, 21, 70, 2.965183, 21, 71,
3.010052, 22, 1, 2.916967, 22, 2, 2.971719, 22, 3, 3.142532, 22, 4, 3.405692, 22, 5, 4.554161,
22, 6, 3.008795, 22, 7, 3.280683, 22, 8, 3.036748, 22, 9, 3.137719, 22, 10, 3.107202, 22, 11,
3.107826, 22, 12, 3.123321, 22, 13, 4.146996, 22, 14, 3.926766, 22, 15, 2.965987, 22, 16,
2.715799, 22, 17, 5.473082, 22, 18, 3.957633, 22, 19, 2.865595, 22, 20, 3.141001, 22, 21,
3.009667, 22, 22, 3.107318, 22, 23, 3.052465, 22, 24, 2.766068, 22, 25, 3.109204, 22, 26,
3.072103, 22, 27, 2.936901, 22, 28, 2.894732, 22, 29, 3.417199, 22, 30, 3.483691, 22, 31,
3.039112, 22, 32, 3.277109, 22, 33, 4.256766, 22, 34, 3.471103, 22, 35, 2.966151, 22, 36,
3.972211, 22, 37, 2.932272, 22, 38, 3.10804, 22, 39, 2.681417, 22, 40, 3.000679, 22, 41, 3.134,
22, 42, 3.073721, 22, 43, 2.929754, 22, 44, 3.194837, 22, 45, 3.036696, 22, 46, 3.379069,
22, 47, 3.003663, 22, 48, 2.895153, 22, 49, 4.276477, 22, 50, 4.363778, 22, 51, 3.926956,
22, 52, 3.719679, 22, 53, 3.574737, 22, 54, 3.107047, 22, 55, 3.333783, 22, 56, 3.830023,
22, 57, 3.022077, 22, 58, 3.152046, 22, 59, 3.054692, 22, 60, 3.10799, 22, 61, 3.067657,

22, 67, 3.135955, 22, 68, 4.198936, 22, 69, 3.055655, 22, 70, 3.462658, 22, 71, 2.894257,
23, 1, 5.448601, 23, 2, 5.341642, 23, 3, 4.927868, 23, 4, 5.171371, 23, 5, 3.036723, 23, 6,
3.588327, 23, 7, 2.938017, 23, 8, 2.84929, 23, 9, 3.269541, 23, 10, 3.10771, 23, 11, 3.815575,
23, 12, 3.906065, 23, 13, 4.342161, 23, 14, 3.911631, 23, 15, 2.728626, 23, 16, 3.268583,
23, 17, 3.382697, 23, 18, 4.441129, 23, 19, 3.010465, 23, 20, 3.048824, 23, 21, 2.75243, 23,
22, 3.378106, 23, 23, 3.008761, 23, 24, 3.259887, 23, 25, 4.412802, 23, 26, 3.219872, 23,
27, 3.248674, 23, 28, 3.524697, 23, 29, 3.010291, 23, 30, 3.211903, 23, 31, 4.115753, 23,
32, 4.06763, 23, 33, 3.675759, 23, 34, 3.046107, 23, 35, 3.141874, 23, 36, 3.328454, 23, 37,
3.878633, 23, 38, 3.975097, 23, 39, 2.989921, 23, 40, 3.567122, 23, 41, 2.82323, 23, 42,
2.965935, 23, 43, 4.937708, 23, 44, 3.55778, 23, 45, 2.965688, 23, 46, 3.916637, 23, 47,
3.037808, 23, 48, 3.004675, 23, 49, 3.546679, 23, 50, 3.249837, 23, 51, 3.416174, 23, 52,
2.712932, 23, 53, 2.865882, 23, 54, 2.965672, 23, 55, 3.036717, 23, 56, 4.137554, 23, 57,
2.980128, 23, 58, 2.965626, 23, 59, 2.835839, 23, 60, 3.063724, 23, 61, 3.065672, 23, 62,
4.306517, 23, 63, 3.463804, 23, 64, 3.036436, 23, 65, 2.898328, 23, 66, 2.82752, 23, 67,
3.613665, 23, 68, 3.009961, 23, 69, 3.751622, 23, 70, 2.89443, 24, 1, 4.198046, 24, 2, 3.095083,
24, 3, 3.034761, 24, 4, 3.137668, 24, 5, 2.96578, 24, 6, 3.075065, 24, 7, 3.197883, 24, 8,
3.407744, 24, 9, 5.5148, 24, 10, 3.00966, 24, 11, 2.783041, 24, 12, 3.392031, 24, 13, 3.033781,
24, 14, 3.664779, 24, 15, 3.463721, 24, 16, 3.405004, 24, 17, 3.452487, 24, 18, 2.938006,
24, 19, 3.569153, 24, 20, 3.244845, 24, 21, 3.201716, 24, 22, 2.85977, 24, 23, 3.797721, 24,
24, 4.398807, 24, 25, 3.972535, 24, 26, 3.33874, 24, 27, 3.807796, 24, 28, 2.852094, 24, 29,
3.153089, 24, 30, 4.528644, 24, 31, 3.412689, 24, 32, 2.933936, 24, 33, 4.72777, 24, 34,
2.964978, 24, 35, 3.034908, 24, 36, 4.453248, 24, 37, 3.062693, 24, 38, 2.761698, 24, 39,
3.057888, 24, 40, 5.037656, 24, 41, 3.391854, 24, 42, 3.901726, 24, 43, 2.90679, 24, 44,
3.046792, 24, 45, 3.415796, 24, 46, 3.039019, 24, 47, 3.395743, 24, 48, 3.469002, 24, 49,
3.016404, 24, 50, 3.386775, 24, 51, 3.517416, 24, 52, 4.710701, 24, 53, 2.965861, 24, 54,
3.754352, 24, 55, 2.832828, 24, 56, 2.82411, 24, 57, 3.141663, 24, 58, 2.775062, 24, 59,
4.330683, 24, 60, 3.994398, 24, 61, 4.140464, 24, 62, 3.218758, 24, 63, 3.036697, 24, 64,
2.89443, 24, 65, 3.587998, 24, 66, 3.614708, 24, 67, 3.178786, 24, 68, 3.604984, 24, 69,
3.492701, 24, 70, 3.239742, 24, 71, 2.82401, 25, 1, 4.852298, 25, 2, 3.248778, 25, 3, 4.078742,
25, 4, 3.566756, 25, 5, 3.036763, 25, 6, 3.406207, 25, 7, 5.211879, 25, 8, 3.178823, 25, 9,
3.038253, 25, 10, 2.923652, 25, 11, 2.754086, 25, 12, 3.916219, 25, 13, 2.745223, 25, 14,
2.966753, 25, 15, 3.002466, 25, 16, 2.937827, 25, 17, 3.041712, 25, 18, 3.281695, 25, 19,
3.24177, 25, 20, 2.96502, 25, 21, 3.009933, 25, 22, 2.894714, 25, 23, 3.178871, 25, 24, 2.82617,
25, 25, 3.805667, 25, 26, 3.15468, 25, 27, 4.306536, 25, 28, 3.009711, 25, 29, 3.108667, 25,
30, 3.073689, 25, 31, 3.249056, 25, 32, 3.283704, 25, 33, 3.11406, 25, 34, 3.143416, 25, 35,
2.823798, 25, 36, 2.964906, 25, 37, 3.542491, 25, 38, 3.55675, 25, 39, 2.866163, 25, 40,
3.507709, 25, 41, 2.696526, 25, 42, 3.421792, 25, 43, 3.036858, 25, 44, 3.081739, 25, 45,
3.14672, 25, 46, 3.366356, 25, 47, 3.007628, 25, 48, 2.765292, 25, 49, 4.169335, 25, 50,
3.665873, 25, 51, 3.227669, 25, 52, 3.090726, 25, 53, 3.263819, 25, 54, 3.065687, 25, 55,
4.81862, 25, 56, 2.960178, 25, 57, 4.380679, 25, 58, 3.394775, 25, 59, 2.968079, 25, 60, 3.1077,
25, 61, 4.843667, 25, 62, 4.775718, 25, 63, 2.780331, 25, 64, 3.135768, 25, 65, 3.21022, 25,
66, 4.713289, 25, 67, 3.036375, 25, 68, 2.864865, 25, 69, 3.5137, 25, 70, 3.237095, 25, 71,
3.10783, 26, 1, 2.823182, 26, 2, 3.337652, 26, 3, 3.395969, 26, 4, 4.323972, 26, 5, 2.997808,
26, 6, 2.965053, 26, 7, 3.526184, 26, 8, 2.904503, 26, 9, 2.965737, 26, 10, 3.196523, 26,
11, 3.575747, 26, 12, 3.047155, 26, 13, 3.160203, 26, 14, 3.060718, 26, 15, 5.001542, 26,
16, 3.06068, 26, 17, 2.937732, 26, 18, 3.010158, 26, 19, 3.107694, 26, 20, 3.107698, 26, 21,
3.225357, 26, 22, 3.44022, 26, 23, 2.954161, 26, 24, 3.247796, 26, 25, 2.966125, 26, 26,
2.681923, 26, 27, 3.10871, 26, 28, 2.909058, 26, 29, 2.942739, 26, 30, 3.314685, 26, 31,
2.792077, 26, 32, 2.835004, 26, 33, 2.965379, 26, 34, 2.681811, 26, 35, 3.128848, 26, 36,
3.112316, 26, 37, 3.00072, 26, 38, 3.199699, 26, 39, 3.700707, 26, 40, 4.290875, 26, 41,
3.134081, 26, 42, 2.68313, 26, 43, 2.958744, 26, 44, 3.293679, 26, 45, 2.848535, 26, 46,
3.009831, 26, 47, 2.967155, 26, 48, 2.823422, 26, 49, 2.865413, 26, 50, 3.612949, 26, 51,
3.148922, 26, 52, 3.495692, 26, 53, 2.980339, 26, 54, 3.101998, 26, 55, 3.039678, 26, 56,
3.470832, 26, 57, 4.708715, 26, 58, 2.772371, 26, 59, 2.976624, 26, 60, 3.164746, 26, 61,
3.464714, 26, 62, 2.919064, 26, 63, 3.27307, 26, 64, 3.321017, 26, 65, 3.626363, 26, 66,
3.605417, 26, 67, 2.894244, 26, 68, 3.4131, 26, 69, 4.239107, 26, 70, 3.0364, 26, 71, 2.895072,
26, 72, 3.792672, 26, 73, 3.03668, 27, 1, 3.211378, 27, 2, 3.182769, 27, 3, 4.098727, 27, 4,
2.828143, 27, 5, 2.851143, 27, 6, 3.306204, 27, 7, 3.010033, 27, 8, 3.144123, 27, 9, 3.186977,
27, 10, 2.920076, 27, 11, 3.132176, 27, 12, 3.255059, 27, 13, 3.332721, 27, 14, 2.938239,
27, 15, 3.907699, 27, 16, 2.650592, 27, 17, 2.689569, 27, 18, 2.998891, 27, 19, 5.49267, 27,
20, 3.006712, 27, 21, 3.247761, 27, 22, 3.084144, 27, 23, 2.921273, 27, 24, 2.839285, 27,
25, 3.404193, 27, 26, 3.164096, 27, 27, 5.49187, 27, 28, 2.683984, 27, 29, 3.564407, 27,
30, 3.67792, 27, 31, 3.036851, 27, 32, 3.253916, 27, 33, 3.297622, 27, 34, 3.993436, 27,
35, 3.491661, 27, 36, 3.710939, 27, 37, 3.107988, 27, 38, 3.406961, 27, 39, 3.615801, 27,
40, 3.226767, 27, 41, 2.835212, 27, 42, 3.155738, 27, 43, 3.492733, 27, 44, 2.703873, 27,
45, 3.153701, 27, 46, 4.784099, 27, 47, 2.683456, 27, 48, 2.904144, 27, 49, 3.147788, 27,
50, 2.896731, 27, 51, 3.081744, 27, 52, 3.600607, 27, 53, 3.296405, 27, 54, 2.965876, 27,
55, 2.857594, 27, 56, 2.823682, 27, 57, 3.127257, 27, 58, 3.230263, 27, 59, 3.105136, 27,
60, 2.783792, 27, 61, 2.968722, 27, 62, 2.895206, 27, 63, 2.965746, 27, 64, 3.036423, 27,
65, 3.780098, 27, 66, 2.929219, 27, 67, 2.911446, 27, 68, 3.252665, 27, 69, 2.896979, 27,
70, 3.389119, 27, 71, 4.064497, 27, 72, 3.337722, 28, 1, 2.965065, 28, 2, 3.009689, 28, 3,
4.022811, 28, 4, 3.036102, 28, 5, 2.898275, 28, 6, 3.374751, 28, 7, 3.960709, 28, 8, 2.872419,
28, 9, 3.214786, 28, 10, 3.400726, 28, 11, 4.67638, 28, 12, 3.349403, 28, 13, 2.959996, 28,

```
    2.937812, 28, 25, 3.491661, 28, 26, 3.072861, 28, 27, 2.894852, 28, 28, 2.894734, 28, 29,
    4.609131, 28, 30, 3.036852, 28, 31, 2.932985, 28, 32, 3.266304, 28, 33, 2.895826, 28, 34,
    2.843803, 28, 35, 2.93755, 28, 36, 2.941332, 28, 37, 3.31748, 28, 38, 3.075332, 28, 39,
    2.835454, 28, 40, 3.250113, 28, 41, 3.550779, 28, 42, 3.004294, 28, 43, 3.698597, 28, 44,
    2.929654, 28, 45, 2.834977, 28, 46, 2.855872, 28, 47, 3.3188, 28, 48, 3.172811, 28, 49, 2.97,
    28, 50, 2.707311, 28, 51, 2.753276, 28, 52, 3.304833, 28, 53, 2.924013, 28, 54, 3.001992,
    28, 55, 3.628564, 28, 56, 4.07469, 28, 57, 2.980158, 28, 58, 3.216737, 28, 59, 3.291629, 28,
    60, 3.327994, 28, 61, 3.960988, 28, 62, 3.41674, 28, 63, 4.994669, 28, 64, 3.178733, 28,
    65, 3.009789, 28, 66, 4.320185, 28, 67, 3.178715, 28, 68, 3.320074, 28, 69, 2.944418, 28,
    70, 2.937704, 28, 71, 3.239723, 29, 1, 3.385381, 29, 2, 2.901883, 29, 3, 3.040719, 29, 4,
    3.299424, 29, 5, 3.212747, 29, 6, 2.88701, 29, 7, 2.864982, 29, 8, 3.276064, 29, 9, 2.894382,
    29, 10, 2.753274, 29, 11, 3.472221, 29, 12, 5.283871, 29, 13, 2.92858, 29, 14, 3.218133, 29,
    15, 3.205037, 29, 16, 3.159125, 29, 17, 3.46268, 29, 18, 3.273149, 29, 19, 3.072734, 29, 20,
    3.036797, 29, 21, 7.337292, 29, 22, 4.116405, 29, 23, 3.196864, 29, 24, 3.701744, 29, 25,
    3.112028, 29, 26, 2.951081, 29, 27, 3.222875, 29, 28, 3.58486, 29, 29, 5.621195, 29, 30,
    3.167728, 29, 31, 3.783694, 29, 32, 2.937863, 29, 33, 4.247767, 29, 34, 2.830144, 29, 35,
    2.996708, 29, 36, 2.965719, 29, 37, 2.865764, 29, 38, 3.369788, 29, 39, 3.267697, 29, 40,
    3.458176, 29, 41, 3.296727, 29, 42, 5.235661, 29, 43, 3.107716, 29, 44, 3.488775, 29, 45,
    2.895266, 29, 46, 3.75203, 29, 47, 3.735099, 29, 48, 3.422747, 29, 49, 3.039737, 29, 50,
    3.216726, 29, 51, 2.881609, 29, 52, 2.993136, 29, 53, 2.906841, 29, 54, 4.287591, 29, 55,
    3.338757, 29, 56, 2.869418, 29, 57, 5.749616, 29, 58, 3.602333, 29, 59, 3.185791, 29, 60,
    4.247771, 29, 61, 2.894765, 29, 62, 3.943762, 29, 63, 2.999758, 29, 64, 3.115702, 29, 65,
    2.836326, 29, 66, 3.123712, 29, 67, 3.010131, 29, 68, 2.966202, 29, 69, 3.010222, 29, 70,
    2.6816, 30, 1, 2.792248, 30, 2, 3.009744, 30, 3, 4.279447, 30, 4, 4.424729, 30, 5, 3.067817,
    30, 6, 4.690888, 30, 7, 2.968156, 30, 8, 3.006951, 30, 9, 2.937719, 30, 10, 2.85496, 30,
    11, 2.756089, 30, 12, 3.153154, 30, 13, 3.200607, 30, 14, 5.113596, 30, 15, 3.655108, 30,
    16, 2.895084, 30, 17, 6.23735, 30, 18, 3.393746, 30, 19, 3.036779, 30, 20, 2.896909, 30,
    21, 2.71196, 30, 22, 3.439973, 30, 23, 3.024728, 30, 24, 3.699007, 30, 25, 3.38308, 30, 26,
    3.432045, 30, 27, 2.90499, 30, 28, 2.987863, 30, 29, 3.326932, 30, 30, 3.443797, 30, 31,
    3.107703, 30, 32, 3.081724, 30, 33, 2.823659, 30, 34, 3.18611, 30, 35, 2.965782, 30, 36,
    3.040714, 30, 37, 3.120758, 30, 38, 3.052707, 30, 39, 3.833731, 30, 40, 2.846, 30, 41, 3.520489,
    30, 42, 2.96598, 30, 43, 3.144161, 30, 44, 3.927763, 30, 45, 3.837581, 30, 46, 4.046637,
    30, 47, 7.372662, 30, 48, 3.476288, 30, 49, 3.316771, 30, 50, 3.296664, 30, 51, 2.752644,
    30, 52, 3.17847, 30, 53, 3.086287, 30, 54, 3.184791, 30, 55, 3.272803, 30, 56, 3.08186, 30,
    57, 2.825634, 30, 58, 2.846113, 30, 59, 3.109416, 30, 60, 2.965819, 30, 61, 2.752005, 30,
    62, 3.063842, 30, 63, 3.125732, 30, 64, 3.226041, 30, 65, 3.107773, 30, 66, 3.154154, 30,
    67, 3.037766, 30, 68, 3.657803, 30, 69, 2.859695, 30, 70, 3.053847, 30, 71, 4.030793};
ss256Instance = {1, 600.613317, 83 239 490, 2104.61, 2359.632257, 2, 600.802977, 83 124 526, 2216.84,
    2367.369083, 3, 600.951366, 82 777 959, 2263.04, 2363.240717, 4, 600.892459, 82 837 373, 2167.06,
    2369.561748, 5, 600.833055, 82 741 270, 2197.6, 2367.522061, 6, 600.827558, 82 268 595, 2246.2,
    2363.056741, 7, 600.888872, 82 235 802, 2121.46, 2363.140726, 8, 601.069034, 82 431 669, 2311.39,
    2373.661133, 9, 600.677222, 81 811 976, 2193.09, 2353.145248, 10, 600.623846, 82 284 976, 2262.85,
    2365.131416, 11, 600.922988, 82 392 656, 2253.38, 2373.62114, 12, 600.954167, 82 333 496, 2156.83,
    2369.348787, 13, 600.303049, 82 136 930, 2164.05, 2370.589597, 14, 600.317735, 82 141 086,
    2291.18, 2364.487524, 15, 600.812894, 72 356 079, 2581.92, 2365.206405, 16, 600.840842,
    70 867 985, 2593.31, 2368.860855, 17, 600.783487, 70 731 198, 2524.09, 2372.574296, 18,
    600.350434, 70 693 961, 2543.25, 2374.432007, 19, 601.013033, 70 734 057, 2586.18, 2369.827709,
    20, 600.815561, 70 706 445, 2570.23, 2369.552757, 21, 600.684237, 70 605 647, 2514.25,
    2368.297937, 22, 600.980615, 70 701 452, 2633.84, 2370.38964, 23, 600.510128, 70 648 540,
    2589.01, 2372.64628, 24, 601.107014, 70 639 005, 2669.77, 2369.910699, 25, 600.64688, 70 467 972,
    2568.93, 2365.841322, 26, 600.712097, 70 576 735, 2641.88, 2370.70658, 27, 601.068203,
    70 535 536, 2651.75, 2366.361238, 28, 600.700146, 70 434 573, 2573.66, 2368.65489, 29,
    600.951965, 70 433 025, 2587.5, 2370.79456, 30, 600.52256, 70 359 152, 2638.73, 2367.004137};
ss32768Instance = {1, 600.206127, 211 542 320, 51.72, 1840.607165, 2, 600.308412, 208 359 397,
    35.71, 1802.592934, 3, 600.217078, 208 572 128, 37.05, 1808.184089, 4, 600.198318, 211 605 452,
    47.05, 1836.375814, 5, 600.238526, 210 904 780, 43.4, 1829.048917, 6, 600.268151, 208 635 903,
    40.77, 1832.986324, 7, 600.19829, 208 837 233, 117.1, 1826.561294, 8, 600.269915, 212 468 478,
    124.43, 1864.127578, 9, 600.259056, 209 218 390, 144.43, 1828.604999, 10, 600.25501, 212 284 435,
    133.71, 1856.661718, 11, 600.199495, 205 207 992, 163.75, 1789.47194, 12, 600.24163,
    209 756 526, 141.34, 1836.008865, 13, 600.338953, 204 089 967, 135.92, 1775.886002, 14,
    600.21563, 210 133 310, 140.29, 1834.56108, 15, 600.299611, 212 043 041, 161.01, 1858.085511,
    16, 600.30368, 210 911 472, 147.1, 1833.84519, 17, 600.220546, 211 931 391, 137.97, 1850.125716,
    18, 600.219486, 209 189 891, 173.06, 1829.231891, 19, 600.327951, 204 058 779, 147.56,
    1780.5033, 20, 600.283519, 207 290 329, 171.83, 1811.065654, 21, 600.367756, 211 357 133,
    138.46, 1847.448123, 22, 600.330594, 209 192 828, 126.65, 1824.136661, 23, 600.24343,
    204 052 041, 196.53, 1783.83879, 24, 600.142599, 210 698 933, 174.82, 1838.62546, 25, 600.338215,
    213 396 292, 150.54, 1863.878634, 26, 600.15324, 207 583 528, 132.69, 1798.983493, 27,
    600.211264, 205 010 225, 143.1, 1791.954562, 28, 600.266739, 208 820 160, 138.54, 1815.79593, 29,
    600.202913, 207 489 694, 154.96, 1801.477112, 30, 600.218, 209 692 043, 141.15, 1824.950543};
```

SQL Statement Elapsed Times

There are two sample sets: 256 CBC latches and 32768 CBC latches. Within each sample set there are 30 ten minute periods where a non-fixed number of elapsed time samples where colletected using the OraPub Elapsed Time Sampler.

With 256 CBC Latches

In[5]:=

```
ss = ss256Elapsed;
masterSamples = 30;

ncols = 3;
masterSampleCol = 1;
elapsedSampleCol = 2;
elapsedTimeCol = 3;
loops = Length[ss] / ncols;

For[i = 1, i ≤ masterSamples, i++,
  sampleSetE[i] = {};
 ];

For[i = 0, i < loops, i++,
  masterNo = ss[[ncols i + masterSampleCol]];
  sqlNo = ss[[ncols i + elapsedSampleCol]];
  elapsedTime = ss[[ncols i + elapsedTimeCol]];
  (*Print["i=",i," masterNo=",masterNo," sqlNo=",sqlNo," elaspedTime=",elapsedTime];*)
  AppendTo[sampleSetE[masterNo], elapsedTime];
 ];
sampleSetE[1];
Mean[sampleSetE[1]];

sampleSetMeans = {};
For[i = 1, i ≤ masterSamples, i++,
  AppendTo[sampleSetMeans, Mean[sampleSetE[i]]];
 ];
sampleSetMeans256 = sampleSetMeans
Mean[sampleSetMeans256]
```

Out[18]=

```
{20.0405, 20.4399, 19.9489, 20.4586, 20.2505, 19.9633, 20.6613, 21.2576, 20.4169, 20.5546,
 20.6866, 20.4881, 19.9484, 21.0969, 25.6916, 25.3674, 25.8227, 25.0741, 26.2919, 25.534,
 25.8392, 25.6045, 24.9854, 26.2543, 25.7559, 25.9441, 25.6114, 25.1614, 24.7593, 26.6672}
```

Out[19]=

```
23.2192
```

With 32768 CBC Latches

In[20]:=
```
ss = ss32768Elapsed;
masterSamples = 30;

ncols = 3;
masterSampleCol = 1;
elapsedSampleCol = 2;
elapsedTimeCol = 3;
loops = Length[ss] / ncols;

For[i = 1, i ≤ masterSamples, i++,
  sampleSetE[i] = {};
 ];

For[i = 0, i < loops, i++,
  masterNo = ss[[ncols i + masterSampleCol]];
  sqlNo = ss[[ncols i + elapsedSampleCol]];
  elapsedTime = ss[[ncols i + elapsedTimeCol]];
  (*Print["i=",i," masterNo=",masterNo," sqlNo=",sqlNo," elaspedTime=",elapsedTime];*)
  AppendTo[sampleSetE[masterNo], elapsedTime];
 ];
sampleSetE[1];
Mean[sampleSetE[1]];

sampleSetMeans = {};
For[i = 1, i ≤ masterSamples, i++,
  AppendTo[sampleSetMeans, Mean[sampleSetE[i]]];
 ];
sampleSetMeans32768 = sampleSetMeans
Mean[sampleSetMeans32768]
```

Out[33]=
```
{3.351, 3.2718, 3.42482, 3.31924, 3.31372, 3.3013, 3.28903, 3.30025, 3.48685, 3.33321,
 3.33563, 3.34116, 3.35855, 3.40569, 3.44219, 3.34184, 3.40581, 3.48969, 3.39749, 3.43376,
 3.37859, 3.32419, 3.46004, 3.43087, 3.34747, 3.2134, 3.25587, 3.26886, 3.41713, 3.35299}
```

Out[34]=
```
3.35975
```

Comparing two situations Numerically

In[35]:=
```
myData = {
   {256, Mean[sampleSetMeans256], 0, Median[sampleSetMeans256], 0, Length[sampleSetMeans256]},
   {32 768, Mean[sampleSetMeans32768],
    100 * (Mean[sampleSetMeans32768] - Mean[sampleSetMeans256]) / Mean[sampleSetMeans256],
    Median[sampleSetMeans32768], 100 * (Median[sampleSetMeans32768] - Median[sampleSetMeans256]) /
      Median[sampleSetMeans256],
    Length[sampleSetMeans32768]}
  };
toGrid = Prepend[myData,
   {"CBC\nlatches", "Mean E\n(sec)", "%\nChange", "Median E\n(sec)", "%\nChange", "Samples"}];
Grid[toGrid, Frame → All]
```

Out[37]=

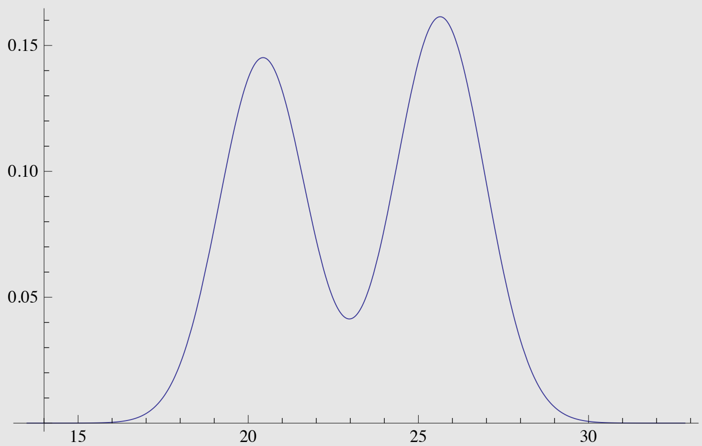
CBC latches	Mean E (sec)	% Change	Median E (sec)	% Change	Samples
256	23.2192	0	24.8723	0	30
32 768	3.35975	-85.5303	3.34924	-86.5343	30

Comparing two situations Graphically

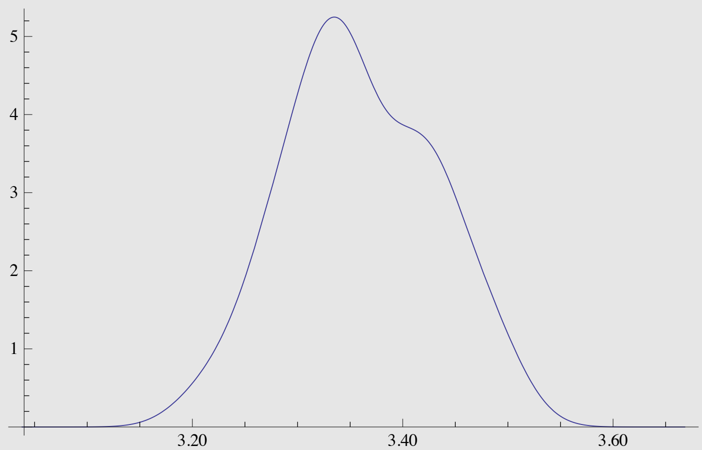
In[38]:=

```
SmoothHistogram[sampleSetMeans256]
SmoothHistogram[sampleSetMeans32768]
SmoothHistogram[{sampleSetMeans256, sampleSetMeans32768}]
```

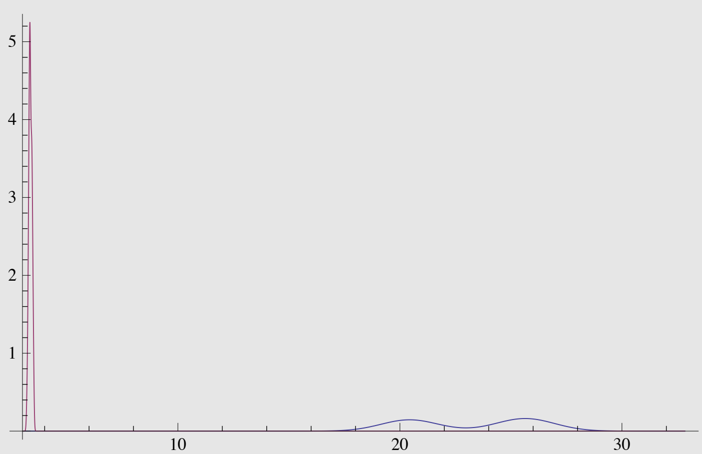
Out[38]=



Out[39]=



Out[40]=



Comparing two situations via Significance Testing

In[41]:=

```
Test1 = MannWhitneyTest[{sampleSetMeans256, sampleSetMeans32768}];
Test2 = LocationEquivalenceTest[
   {sampleSetMeans256, sampleSetMeans32768}, {"TestDataTable", "AutomaticTest"}];
Print["Sig test between 256 latches (", Length[sampleSetMeans256],
  " values) and 32768 latches (", Length[sampleSetMeans32768], " values)."];
Print["P-Values: Test1=", Test1, " Test2=", Test2];
```

```
Sig test between 256 latches (30 values) and 32768 latches (30 values).
```

P-Values: Test1=3.01986×10^{-11} Test2={

	Statistic	P-Value
Kruskal-Wallis	44.2623	4.05641×10^{-19}

, KruskalWallis}

Instance Data; St, Qt, Rt, Arrival Rate

Data Loading

In[45]:=

```
ssNum = 2;
sampleNum = 30;

Clear[ss];
ss[1] = ss256Instance; latches[1] = 256;
ss[2]  = ss32768Instance; latches[2] = 32 768;

ncols = 5;
sampleCol = 1;
elapsedTCol = 2;
lioCol = 3;
waitTCol = 4;
cpuTCol = 5;

Do[
  ssLio[ssidx] = {}; ssL[ssidx] = {}; ssSt[ssidx] = {}; ssQt[ssidx] = {}; ssRt[ssidx] = {};
  theSS = ss[ssidx];
  Table[
   elapsedT = theSS[[ncols sampleidx + elapsedTCol]];
   lioTot = theSS[[ncols sampleidx + lioCol]];
   cpuSecTot = theSS[[ncols sampleidx + cpuTCol]];
   waitSecTot = theSS[[ncols sampleidx + waitTCol]];

   (*Print["sampleidx=",sampleidx," ssidx=",ssidx," elapsedT=",elapsedT," lioTot=",lioTot];*)

   λ = lioTot / (1000 elapsedT);
   St = (cpuSecTot 1000) / lioTot;
   Qt = (waitSecTot 1000) / lioTot;
   Rt = St + Qt;

   AppendTo[ssLio[ssidx], lioTot];
   AppendTo[ssL[ssidx], λ];
   AppendTo[ssSt[ssidx], St];
   AppendTo[ssQt[ssidx], Qt];
   AppendTo[ssRt[ssidx], Rt];

   , {sampleidx, 0, sampleNum - 1}
  ];
  , {ssidx, ssNum}
 ];
Length[ssLio[1]]
Take[ssLio[1], 5]
Mean[ssL[1]]
```

Out[57]=

```
30
```

Out[58]=

```
{83 239 490, 83 124 526, 82 777 959, 82 837 373, 82 741 270}
```

Out[59]=

```
126.851
```

Basic Statistics

In[60]:=

```
myData = Table[
    {
     latches[ssidx], Mean[ssL[ssidx]], Mean[ssSt[ssidx]], Mean[ssQt[ssidx]], Mean[ssRt[ssidx]],
     Length[ssLio[ssidx]], N[StandardDeviation[ssL[ssidx]]], N[StandardDeviation[ssSt[ssidx]]],
     N[StandardDeviation[ssQt[ssidx]]], N[StandardDeviation[ssRt[ssidx]]]
    }, {ssidx, 1, ssNum}
   ];
toGrid = Prepend[myData, {"CBC\nlatches", "Avg L\n(lio/ms)", "Avg St\n(ms/lio)",
    "Avg Qt\n(ms/lio)", "Avg Rt\n(ms/lio)", "Samples", "Stdev L\n(lio/ms)",
    "Stdev St\n(ms/lio)", "Stdev Qt\n(ms/lio)", "Stdev Rt\n(ms/lio)"}];
Grid[
 toGrid,
 Frame →
  All]
```

Out[62]=

CBC latches	Avg L (lio/ms)	Avg St (ms/lio)	Avg Qt (ms/lio)	Avg Rt (ms/lio)	Samples	Stdev L (lio/ms)	Stdev St (ms/lio)	Stdev Qt (ms/lio)	Stdev Rt (ms/lio)
256	126.851	0.0312566	0.0320554	0.063312	30	9.96317	0.00245397	0.00505153	0.00749039
32 768	348.428	0.0087205	0.000605187	0.00932568	30	4.39731	0.0000334309	0.000220609	0.000236183

Highlighting Change

In[121]:=

```
myData = Table[
   {
    latches[ssidx],
    Mean[ssL[ssidx]], NumberForm[100 * (Mean[ssL[ssidx]] - Mean[ssL[1]]) / Mean[ssL[1]], {10, 1}],
    Mean[ssSt[ssidx]], NumberForm[100 * (Mean[ssSt[ssidx]] - Mean[ssSt[1]]) / Mean[ssSt[1]], {10, 1}],
    Mean[ssQt[ssidx]], NumberForm[100 * (Mean[ssQt[ssidx]] - Mean[ssQt[1]]) / Mean[ssQt[1]], {10, 1}],
    Mean[ssRt[ssidx]], NumberForm[100 * (Mean[ssRt[ssidx]] - Mean[ssRt[1]]) / Mean[ssRt[1]], {10, 1}],
    Length[ssLio[ssidx]]
   }, {ssidx, 1, ssNum}
  ];
toGrid = Prepend[myData, {"CBC\nlatches", "Avg L\n(lio/ms)", "%\nChange", "Avg St\n(ms/lio)",
    "%\nChange", "Avg Qt\n(ms/lio)", "%\nChange", "Avg Rt\n(ms/lio)", "%\nChange", "Samples"}];
Grid[toGrid, Frame → All]
myData = Table[
   {
    latches[ssidx],
    Median[ssL[ssidx]],
    NumberForm[100 * (Median[ssL[ssidx]] - Median[ssL[1]]) / Median[ssL[1]], {10, 1}],
    Median[ssSt[ssidx]], NumberForm[100 * (Median[ssSt[ssidx]] - Median[ssSt[1]]) / Median[ssSt[1]],
     {10, 1}],
    Median[ssQt[ssidx]], NumberForm[100 * (Median[ssQt[ssidx]] - Median[ssQt[1]]) / Median[ssQt[1]],
     {10, 1}],
    Median[ssRt[ssidx]], NumberForm[100 * (Median[ssRt[ssidx]] - Median[ssRt[1]]) / Median[ssRt[1]],
     {10, 1}],
    Length[ssLio[ssidx]]
   }, {ssidx, 1, ssNum}
  ];
toGrid = Prepend[myData, {"CBC\nlatches", "Median L\n(lio/ms)", "%\nChange", "Median St\n(ms/lio)",
    "%\nChange", "Median Qt\n(ms/lio)", "%\nChange", "Median Rt\n(ms/lio)", "%\nChange", "Samples"}];
Grid[
 toGrid,
 Frame →
  All]
```

Out[123]=

CBC latches	Avg L (lio/ms)	% Change	Avg St (ms/lio)	% Change	Avg Qt (ms/lio)	% Change	Avg Rt (ms/lio)	% Change	Samples
256	126.851	0.0	0.0312566	0.0	0.0320554	0.0	0.063312	0.0	30
32 768	348.428	174.7	0.0087205	-72.1	0.000605187	-98.1	0.00932568	-85.3	30

Out[126]=

CBC latches	Median L (lio/ms)	% Change	Median St (ms/lio)	% Change	Median Qt (ms/lio)	% Change	Median Rt (ms/lio)	% Change	Samples
256	119.189	0.0	0.0330574	0.0	0.0356466	0.0	0.0687622	0.0	30
32 768	348.535	192.4	0.00872807	-73.6	0.000666802	-98.1	0.00938658	-86.3	30

Instance Focus: Sample Set Normality Tests

Before we can perform a standard t-test hypothesis tests on our data, we need to ensure it is normally distributed...because that is one of the underlying assumptions and requirements for properly performing a t-test.

Statistical and vIsual normality test

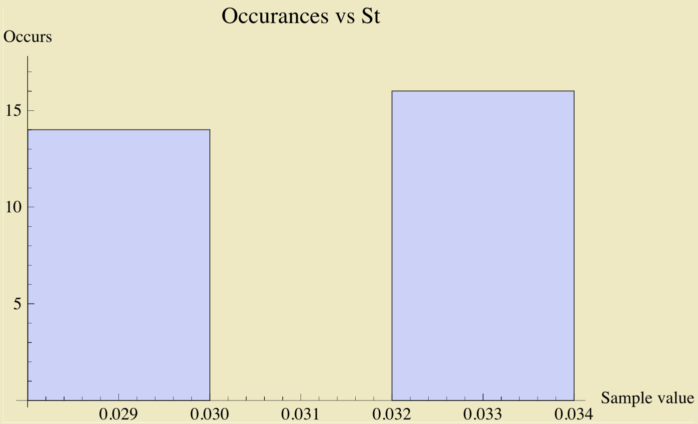
Our alpha will be 0.05, so if the distribution fit test results in a value greater than 0.05 then we can assume the data set is indeed normally distributed.

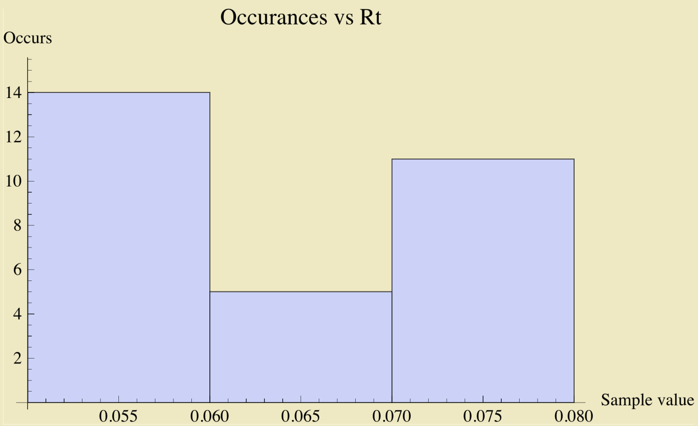
The first test is just to double check to make sure my thinking is correct. Since I creating a normal distribution based on a mean and standard deviation (just happens to be based on the my sample set data), I would expect a p-value (the result) to greatly exceed 0.05. Notice that the more samples I have created (the final number), the closer the p-value approaches 1.0.

In[69]:=

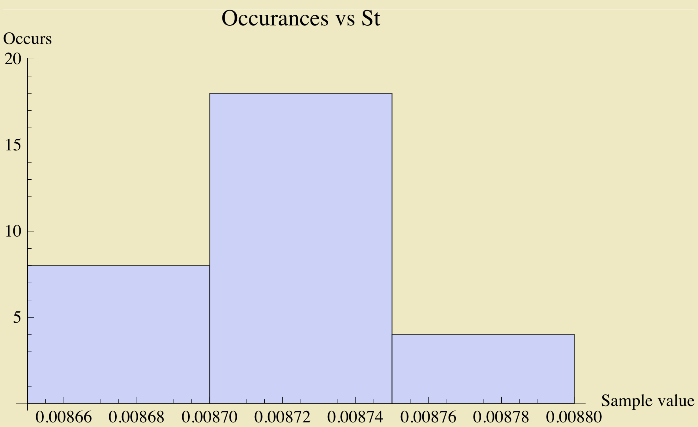
```
check = DistributionFitTest[
   RandomVariate[NormalDistribution[Mean[ssSt[1]], StandardDeviation[ssSt[1]]], 10 000]];
Do[
  pValueSt = DistributionFitTest[ssSt[i]];
  pValueRt = DistributionFitTest[ssRt[i]];
  Print["Sample set ", i, " with ",
   Length[ssSt[i]], " values. P-values: St=", pValueSt, " Rt=", pValueRt];
  st = Histogram[ssSt[i], PlotLabel → "Occurances vs St", AxesLabel → {"Sample value", "Occurs"}];
  rt = Histogram[ssRt[i], PlotLabel → "Occurances vs Rt", AxesLabel → {"Sample value", "Occurs"}];
  Print[st];
  Print[rt];
  Print["----------------------------------------"];
  , {i, 1, ssNum}
 ];
Print["This number should be much greater than 0.05: ",
  check, " If not try again by re-evaluating."];
```

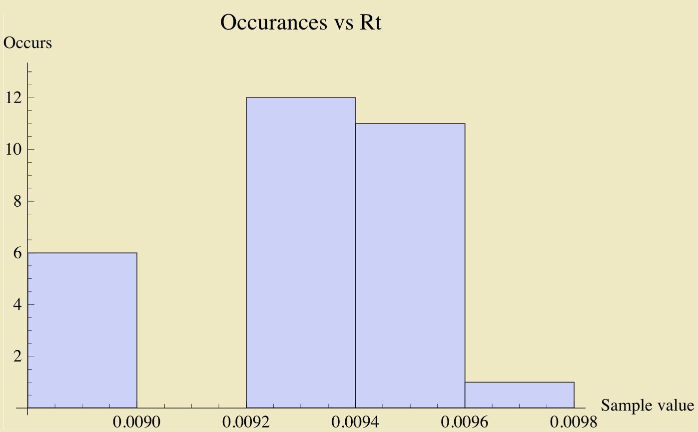
Sample set 1 with 30 values. P-values: St=0. Rt=0.





Sample set 2 with 30 values. P-values: St=0.124093 Rt=3.67931×10^{-6}





This number should be much greater than 0.05: 0.33539 If not try again by re-evaluating.

Instance Focus: Sample Comparison Tests (when normality exists)

Assuming our samples **are normally distributed**, now it's time to see if they are significantly different. If so, then we know changing the number of latches and chains indeed makes a significant performance difference...at least statistically.

The null hypothesis is; there is no real difference between our samples sets. We need to statistically prove that any difference is the result of randomness; like we just happened to pick poor set of samples and it makes their difference look much worse than it really is.

A t-test will produce a statistic p. The p value is a probability, with a value ranging from zero to one. It is the answer to this question: If the populations really have the same mean overall, what is the probability that random sampling would lead to a difference between sample means larger than observed?

For example, if the p value is 0.03 we can say a random sampling from identical populations would lead to a difference smaller than you observed in 97% of the experiments and larger than you observed in 3% of the experiments.

Said another way, suppose I have a single sample set and I copy it, resultling in two identical sample sets. Now suppose we perform a significance test on these two identical sample sets. The resuting p-value will be 1.0 because they are exactly the same. We are essentially doing the same thing here except we have to different sample sets... but we still want to see if they “like” each other..and in our case we hope they are NOT the like each other, which means the p-value will low... below our cut off value of 0.05.

For our analysis we choose alpha of 0.05. To accept that our two samples are statistically similar the p value would need to be less than 0.05 (our alpha).

Good reference about the P-Value and significance testing: http://www.graphpad.com/articles/pvalue.htm

Here we go (assuming our samples are normally distributed):

1. Our P value threshold is 0.05, which is our alpha.
2. The null hypothesis is the two populations have the same mean. (Remember we have to sample sets, which not the population.)
3. Do the statistical test to compute the P value.
4. Compare the result P value to our threshold alpha value. If the P value is less then our threshold, we will reject the null hypothesis and say the difference between our samples is significant. However, if the P value is greater than the threshold, we cannot reject the null hypothesis and any difference between our samples are not statistically significant.

In[72]:=

```
Do[
  pValueSt = TTest[{ssSt[i], ssSt[i + 1]}];
  Print["St: (", Length[ssSt[i]],
   " values) pvalue between sample set ", i, " and ", i + 1, " is ", pValueSt];
  pValueRt = TTest[{ssRt[i], ssRt[i + 1]}];
  Print["Rt: (", Length[ssRt[i]],
   " values) pvalue between sample set ", i, " and ", i + 1, " is ", pValueRt];
  ,
  {i, 1, ssNum - 1}
 ];
```

TTest::nortst : At least one of the p-values in {0., 0.124093}, resulting from a test for normality, is below 0.025`. The tests in {T} require that the data is normally distributed. ≫

St: (30 values) pvalue between sample set 1 and 2 is 8.90612×10^{-30}

TTest::nortst : At least one of the p-values in $\{0., 3.67931 \times 10^{-6}\}$, resulting from a test for normality, is below 0.025`. The tests in {T} require that the data is normally distributed. ≫

Rt: (30 values) pvalue between sample set 1 and 2 is 8.57989×10^{-27}

If the above T-Test results (p value) are less then our threshold we can say there is a significant difference between the two sample sets.

Instance Focus: Sample Comparison Tests (when normality does NOT exist)

If our sample sets are **not normally distributed,** we can not perform a simple t-test. We can perform what are called location tests. I did some research on significance testing when non-normal distributions exists. I found a very nice reference:

http://www.statsoft.com/textbook/nonparametric-statistics

The paragraph below (which is from the reference above) is a key reference to what we're doing here:

...the need is evident for statistical procedures that enable us to process data of "low quality," from small samples, on variables about which nothing is known (concerning their distribution). Specifically, nonparametric methods were developed to be used in cases when the researcher knows nothing about the parameters of the variable of interest in the population (hence the name nonparametric). In more technical terms, nonparametric methods do not rely on the estimation of parameters (such as the mean or the standard deviation) describing the distribution of the variable of interest in the population. Therefore, these methods are also sometimes (and more appropriately) called parameter-free methods or distribution-free methods.

Being that I'm not a statistician but still need to determine if these sample sets are significant different, I let *Mathematica* determine the appropriate test. Notice that one of the above mentioned tests will probably be the test *Mathematica* chooses.

Note: If we run our normally distributed data through this analysis (speically, the "LocationEquivalenceTest"), *Mathematica* should detect this and use a more appropriate significant test, like a t-test.

Here we go with the hypothesis testing (assuming our sample sets are not normally distributed):

1. Our P value threshold is 0.05, which is our alpha.
2. The null hypotheses is the two populations have the same mean. (Remember we have to sample sets, which is not the population.)
3. Do the statistical test to compute the P value.
4. Compare the result P value to our threshold alpha value. If the P value is less then our threshold, we will reject the null hypothesis and say the difference between our samples is significant. (Which is what I'm hoping to see.) However, if the P value is greater than the threshold, we cannot reject the null hypothesis and any difference between our samples are not statistically significant; randomness, picked the "wrong" samples, etc.

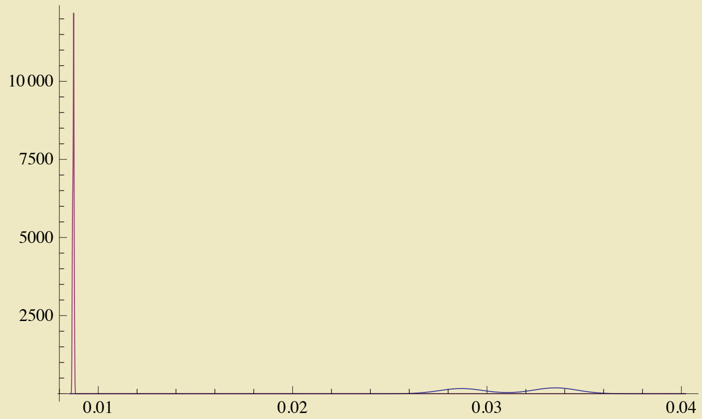
In[73]:=

```
Do[
  StHist = SmoothHistogram[{ssSt[i], ssSt[i + 1]}];
  StTest1 = MannWhitneyTest[{ssSt[i], ssSt[i + 1]}];
  StTest2 = LocationEquivalenceTest[{ssSt[i], ssSt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["St: (", Length[ssSt[i]], " values) Between sample ",
   i, " and ", i + 1, ". Test1=", StTest1, " Test2=", StTest2];
  Print[StHist];
  Print["--------------------------------------------------------"];
  QtHist = SmoothHistogram[{ssQt[i], ssQt[i + 1]}];
  QtTest1 = MannWhitneyTest[{ssQt[i], ssQt[i + 1]}];
  QtTest2 = LocationEquivalenceTest[{ssQt[i], ssQt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["Qt: (", Length[ssQt[i]], " values) Between sample ",
   i, " and ", i + 1, ". Test1=", QtTest1, " Test2=", QtTest2];
  Print[QtHist];
  Print["--------------------------------------------------------"];
  RtHist = SmoothHistogram[{ssRt[i], ssRt[i + 1]}];
  RtTest1 = MannWhitneyTest[{ssRt[i], ssRt[i + 1]}];
  RtTest2 = LocationEquivalenceTest[{ssRt[i], ssRt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["Rt: (", Length[ssRt[i]], " values) Between sample ",
   i, " and ", i + 1, ". Test1=", RtTest1, " Test2=", RtTest2];
  Print[RtHist];
  Print[
   "-----------------------------------------------------------------------------------------
     ---"];
  , {i, 1, ssNum - 1}
 ];
```

St: (30 values) Between sample 1 and 2. Test1= 3.01986×10^{-11} Test2={

	Statistic	P-Value
Kruskal-Wallis	44.2623	4.05641×10^{-19}

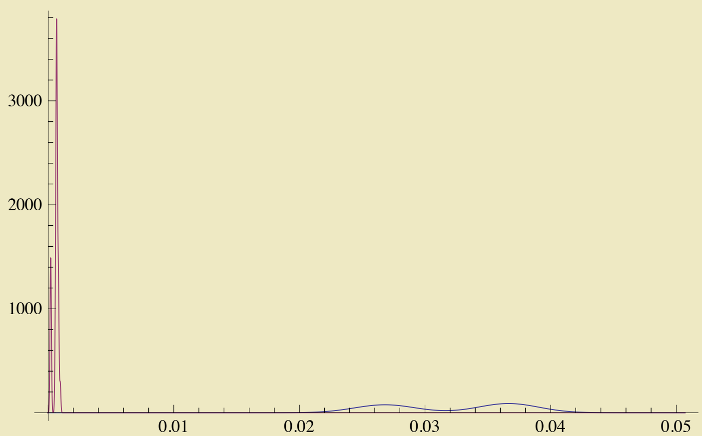
, KruskalWallis}



Qt: (30 values) Between sample 1 and 2. Test1= 3.01986×10^{-11} Test2={

	Statistic	P-Value
Kruskal-Wallis	44.2623	4.05641×10^{-19}

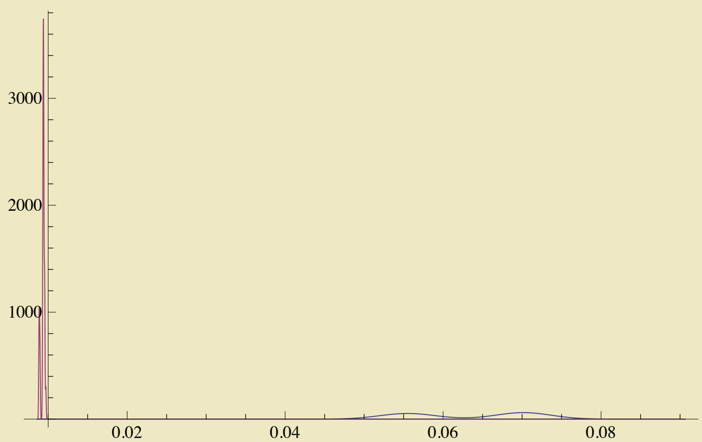
, KruskalWallis}



Rt: (30 values) Between sample 1 and 2. Test1= 3.01986×10^{-11} Test2={

	Statistic	P-Value
Kruskal-Wallis	44.2623	4.05641×10^{-19}

, KruskalWallis}



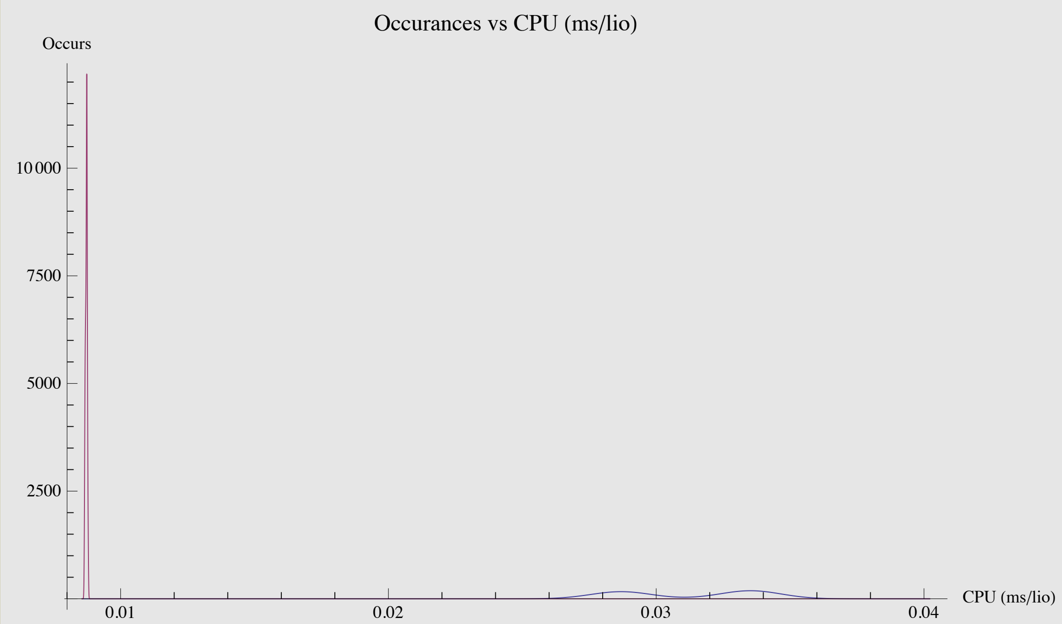
Instance Focus: Visually Comparing All Samples

I also wanted to get a nice visual picture of my sample sets...together.

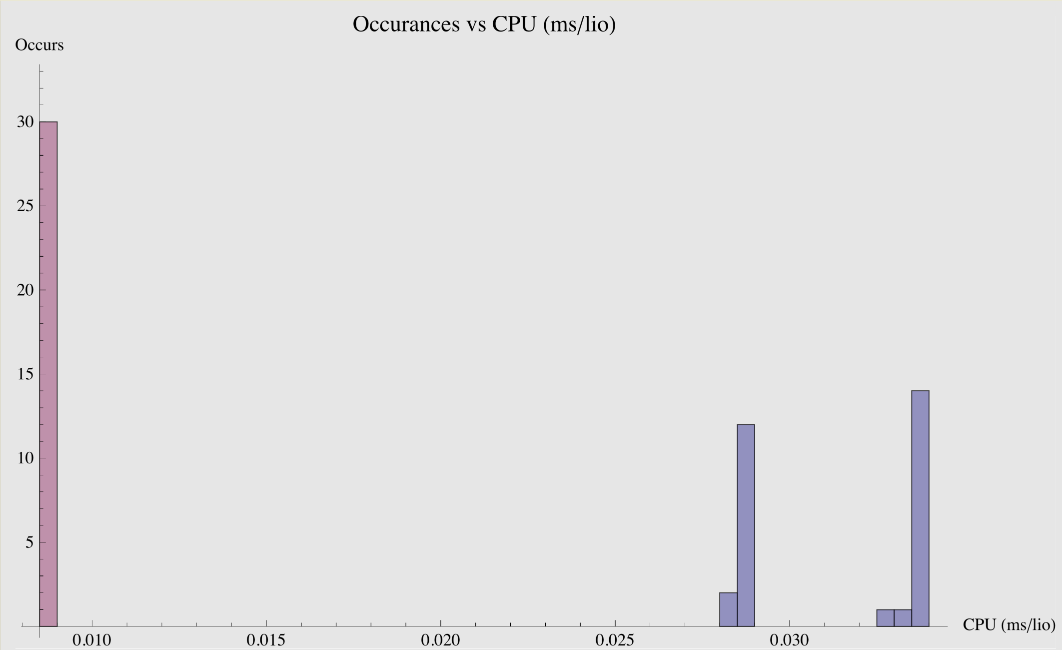
In[74]:=

```
SmoothHistogram[{ssSt[1], ssSt[2]},
 PlotLabel → "Occurances vs CPU (ms/lio)", AxesLabel → {"CPU (ms/lio)", "Occurs"}]
Histogram[{ssSt[1], ssSt[2]}, 40,
 PlotLabel → "Occurances vs CPU (ms/lio)", AxesLabel → {"CPU (ms/lio)", "Occurs"}]
```

Out[74]=



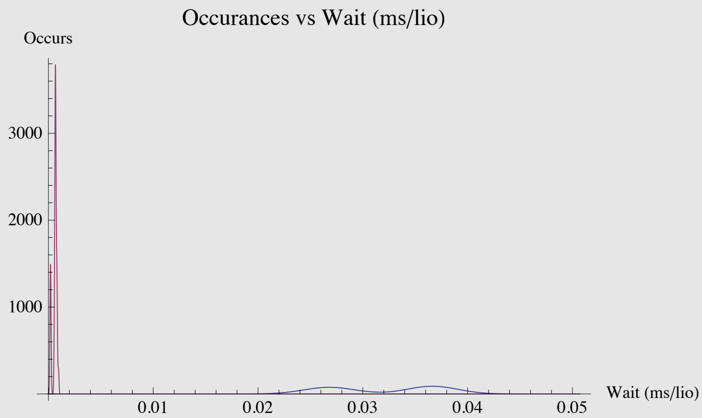
Out[75]=



In[76]:=

```
SmoothHistogram[{ssQt[1], ssQt[2]},
 PlotLabel → "Occurances vs Wait (ms/lio)", AxesLabel → {"Wait (ms/lio)", "Occurs"}]
```

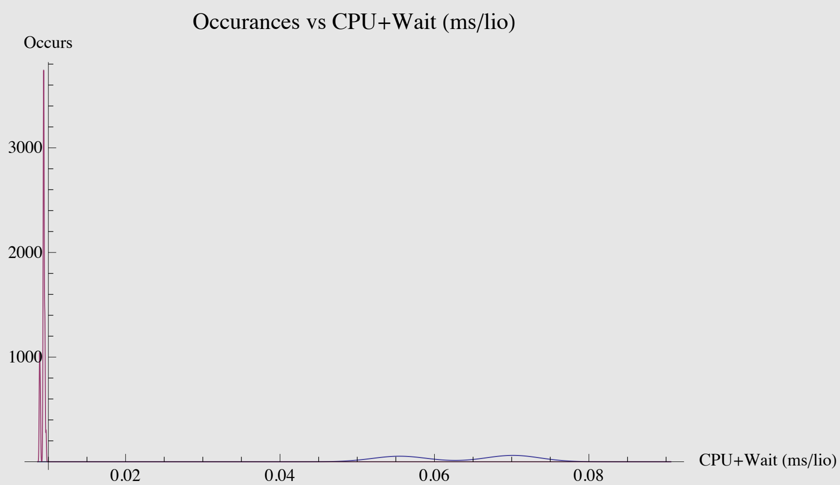
Out[76]=



In[77]:=

```
SmoothHistogram[{ssRt[1], ssRt[2]},
 PlotLabel → "Occurances vs CPU+Wait (ms/lio)", AxesLabel → {"CPU+Wait (ms/lio)", "Occurs"}]
Histogram[{ssRt[1], ssRt[2]}, 40,
 PlotLabel → "Occurances vs CPU+Wait (ms/lio)", AxesLabel → {"CPU+Wait (ms/lio)", "Occurs"}]
```

Out[77]=



Out[78]=