

Question 1a) What compute times do you get for the three inputs (in seconds with all printed digits)? List only the problem size and corresponding compute times.

Problem size: 10000000  
Compute time: 0.007895 s

Problem size: 100000000  
Compute time: 0.093169 s

Problem size: 1000000000  
Compute time: 0.942885 s

Question 1b) Does the compute time increase sub-linearly ( $\text{big-O} < O(n)$ ), linearly ( $\text{big-O} = O(n)$ ), or super-linearly ( $\text{big-O} > O(n)$ ) with respect to the number of elements ( $n$ ) in the vectors (i.e., the command-line parameter)? Explain your answer. Deviations from linear by more than one percent are considered non-linear.

The compute time increases super linearly. This is because the compute time ratio increases more than 1% the ratio of the increase of the problem size. And, the increase in compute time is greater than what would be expected linearly.

Question 2a) What compute times do you get for the three inputs (in seconds with all printed digits)?

Start value: 3  
Upper bound: 2000000  
Compute time: 0.286533 s

Start value: 5  
Upper Bound: 20000000  
Compute time: 3.323377 s

Start value: 7  
Upper Bound: 200000000  
Compute time: 37.838057 s

Question 2b) Does the compute time increase sub-linearly, linearly, or super-linearly with respect to the size of the tested range (i.e., the larger command-line parameter)? Explain your answer. Deviations from linear by more than one percent are considered non-linear.

The compute time increases super linearly. This is because the compute time increases by more than 1% with the change in the input size.

Question 2c) How long is the longest Collatz sequence for the “largest” tested input?

The largest input is a start value of 7 with a an upper bound of 200000000.

Question 2d) What response do you get (include the full message in the report)? Explain why you get this error message.

slurmstepd: error: \*\*\* JOB 4821451 ON c208-021 CANCELLED AT 2022-09-13T10:50:18 DUE TO TIME LIMIT \*\*\*

The collatz sequence will attempt to reach 1 100 million times. It will never reach that amount within the execution time of 30 seconds. Therefore, frontera will automatically terminate the job.

Question 3a) What compute times do you get for the four inputs (in seconds with all printed digits)?

Frames: 32  
Width: 512  
Compute time: 3.404257 s

Frames: 64  
Width: 512  
Compute time: 6.806099 s

Frames: 32  
Width: 1024  
Compute time: 13.567929 s

Frames: 64  
Width: 1024  
Compute time: 27.132434 s

Question 3b) Is the compute time approximately linear or approximately quadratic in the width of the picture (i.e., the first command-line parameter)? Explain your answer.

The compute time change is quadratic with respect to the width. The reasoning for this is the increase in the ratio for input difference versus compute time difference is not equivalent by more than 1%.

Question 3c) Is the compute time linear or super-linear with respect to the number of frames (i.e., the second command-line parameter)? Explain your answer. Deviations from linear by more than a few percent are considered non-linear.

With respect to the frames of the picture, the compute time increases linearly. Each time the frames double, the compute also nearly doubles. The percent difference between the two ratios is less than 1%.

Question 3d) Why are no \*.bmp output files created for any of the four inputs? Check the source code to find the answer.

The program only writes to a BMP file, if the width is less than or equal to 256.

Question 4a) What compute times do you get for the four inputs (in seconds with all printed digits)?

Bodies: 8192  
Time steps: 16  
Compute time: 1.708865 s

Bodies: 8192  
Time steps: 32  
Compute time: 3.419511 s

Bodies: 16384  
Time steps: 16  
Compute time: 6.989605 s

Bodies: 16384  
Time steps: 32  
Compute time: 13.680650 s

Question 4b) Is the runtime approximately linear or approximately quadratic in the number of bodies (i.e., the first command-line parameter)? Explain your answer.

The runtime with respect to the number of bodies is quadratic. This is because the ratio of compute difference is more than 1% greater than the ratio of bodies increase.

Question 4c) Is the runtime approximately linear or approximately quadratic in the number of steps (i.e., the second command-line parameter)? Explain your answer.

The runtime with respect to the number of steps increases quadratically. The reason being the ratio of compute time change is more than 1% different from the ratio of time steps change.

Question 5a) What compute times do you get for the five inputs (in seconds with all printed digits)?

Input: /home1/00976/burtsche/Graphs/USA-road-d.USA.egr  
Nodes: 23947347  
Edges: 57708624  
Compute time: 0.626502 s

Input: /home1/00976/burtsche/Graphs/europe\_osm.egr  
Nodes: 50912018  
Edges: 108109320  
Computed time: 1.147970 s

Input: /home1/00976/burtsche/Graphs/r4-2e23.sym.egr  
Nodes: 8388608  
Edges: 67108846  
Computed time: 0.570491 s

Input: /home1/00976/burtsche/Graphs/soc-LiveJournal1.egr  
Nodes: 4847571  
Edges: 85702474  
Compute time: 0.266633 s

Input: /home1/00976/burtsche/Graphs/uk-2002.egr  
Nodes: 18520486  
Edges: 523574516  
Compute time: 0.745629 s

Question 5b) What percentage of the graph nodes end up in the independent set for each input (list them with one digit after the decimal point)?

Input: /home1/00976/burtsche/Graphs/USA-road-d.USA.egr  
Independent set percentage: 55.0%

Input: /home1/00976/burtsche/Graphs/europe\_osm.egr  
Independent set percentage: 56.5%

Input: /home1/00976/burtsche/Graphs/r4-2e23.sym.egr  
Independent set percentage: 73.7%

Input: /home1/00976/burtsche/Graphs/soc-LiveJournal1.egr  
Independent set percentage: 52.2%

Input: /home1/00976/burtsche/Graphs/uk-2002.egr  
Independent set percentage: 43.0%