

CMPUT379

Assignment3 Report

Zhengyao Zhang (1583946)

Instructions:

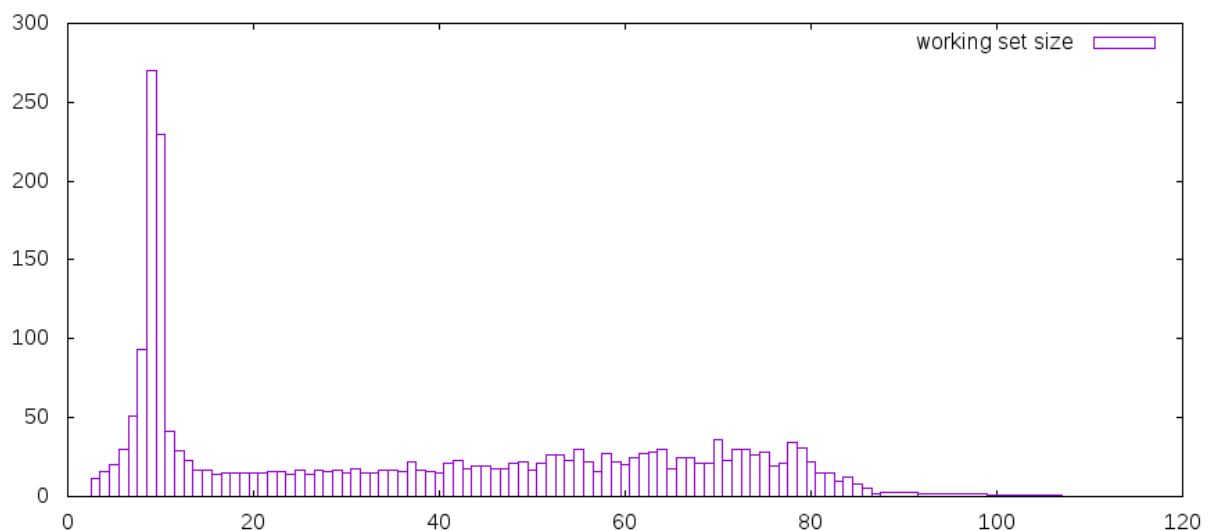
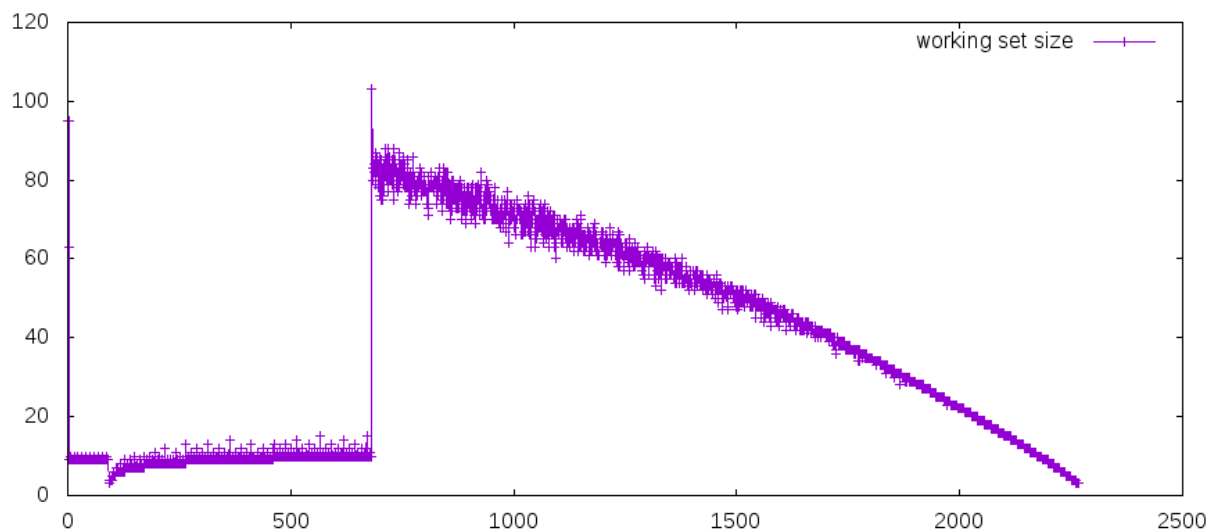
All the necessary files are included in the zip file, basic information of the design of the program is included in 'design.md'. Basically, calling 'make' can generate one executable 'valws379', the purpose of the generated executable and the command line to execute it are included in 'design.md'.

For plotting the data generated by 'valws379', the output need to be redirect to a file named 'data.txt' instead of directly printing to stdout (e.g. `./valws 379 > data.txt`). Then compile the c file 'gnuplot.c' with command line `gcc -o gnuplot gnuplot.c` and execute the generated executable generate two diagrams and report the statistics into stdout.

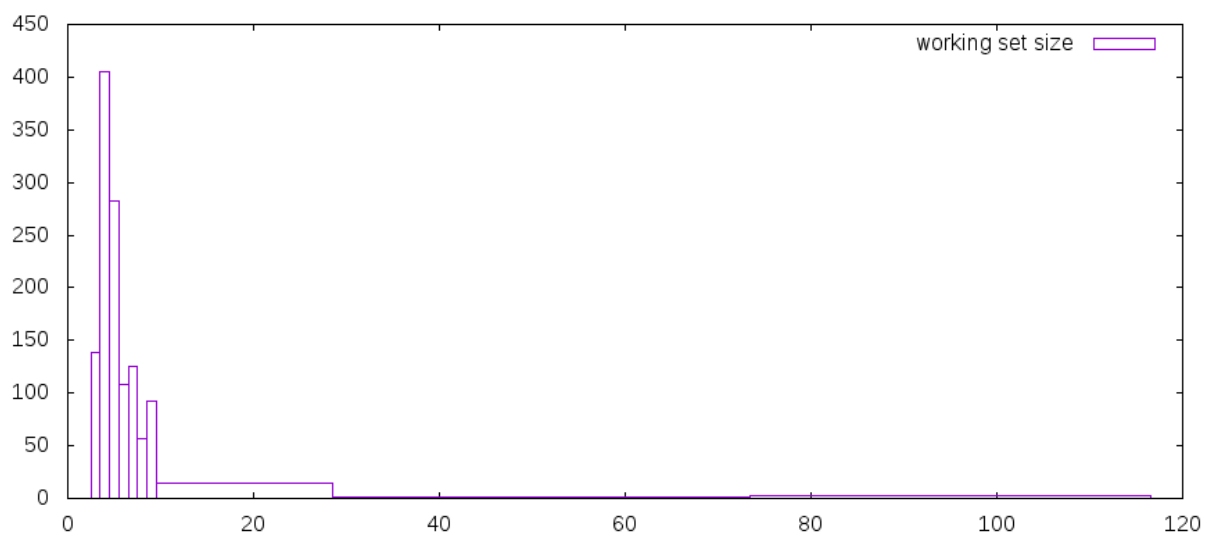
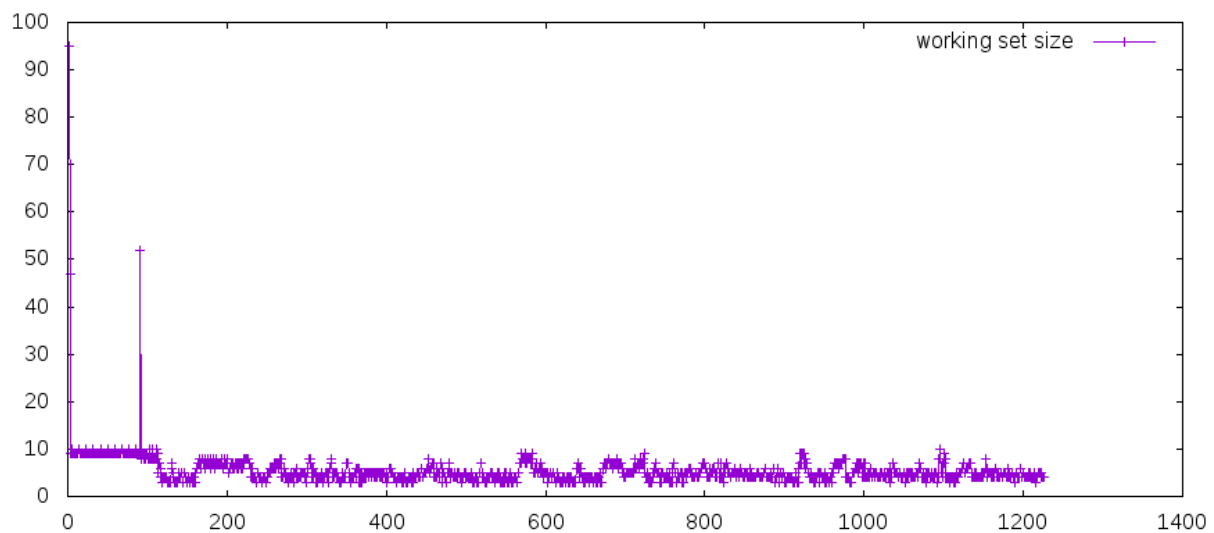
Observations:

(a) Difference between programs:

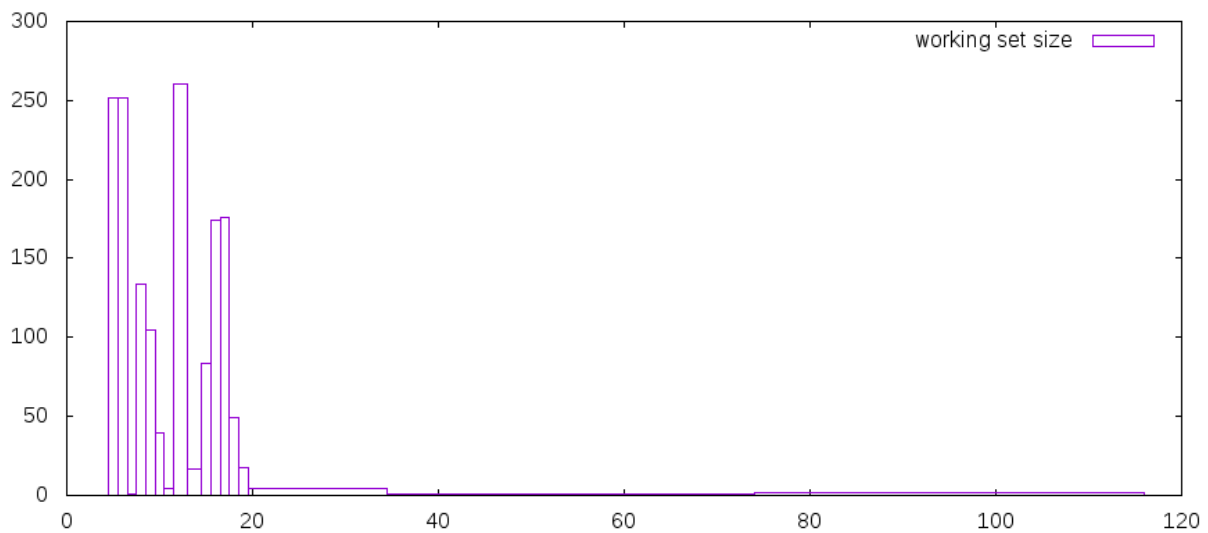
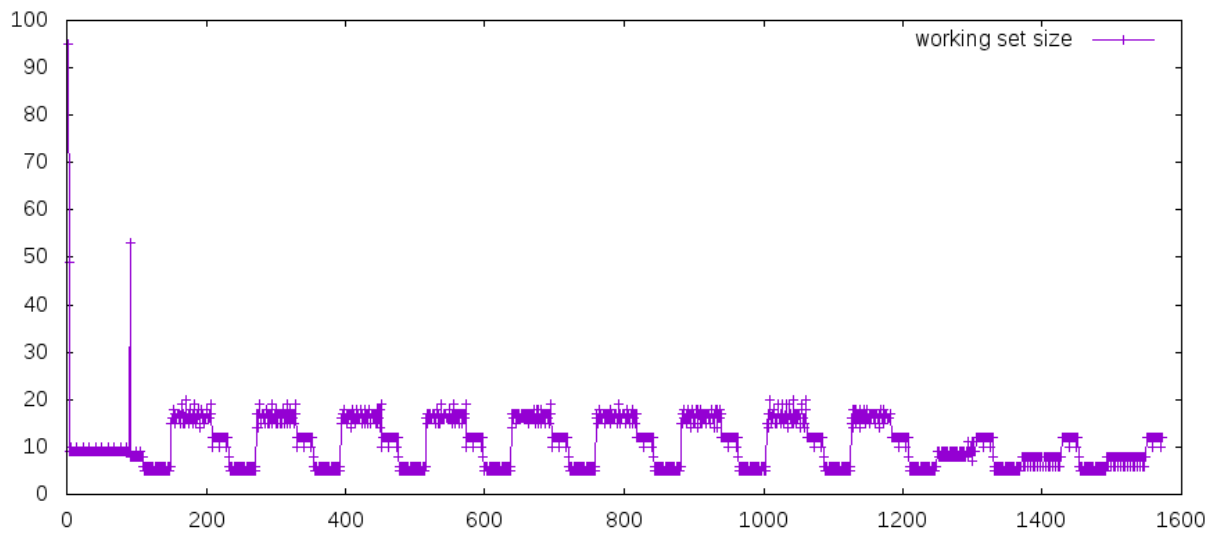
Heapsort 100000 elements, pagesize 4096, window size 100000



Quicksort 100000 elements, pagesize 4096, window size 100000



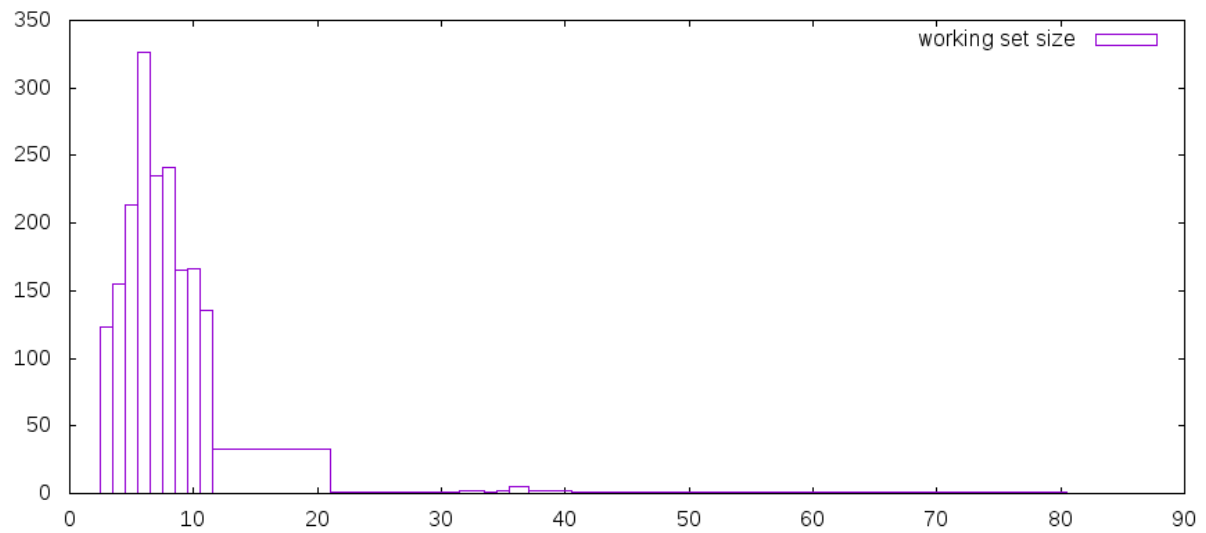
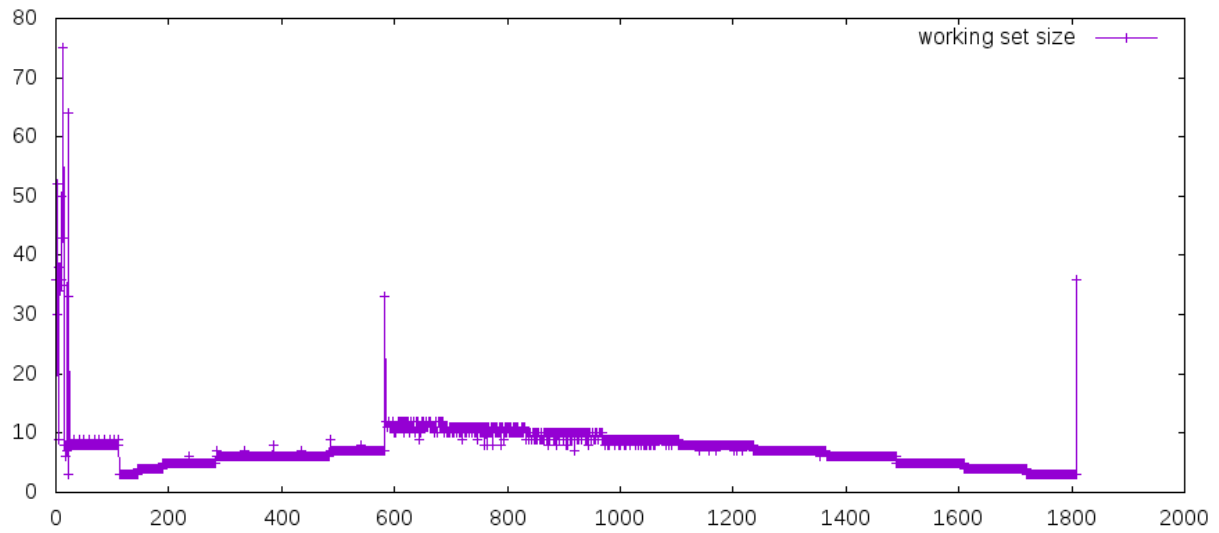
Radixsort 100000 elements, pagesize 4096, window size 100000



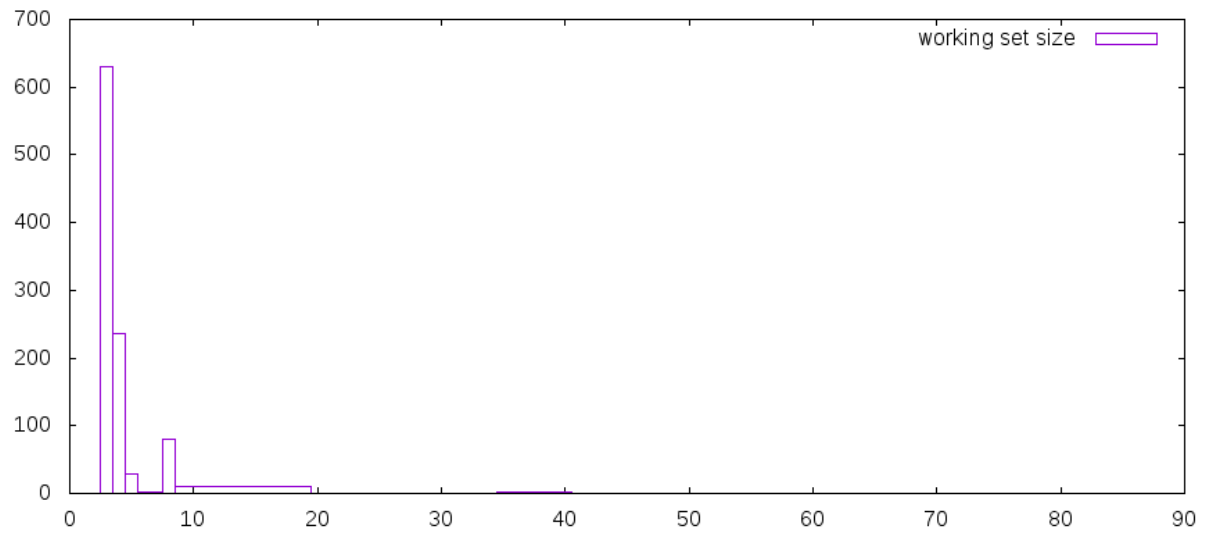
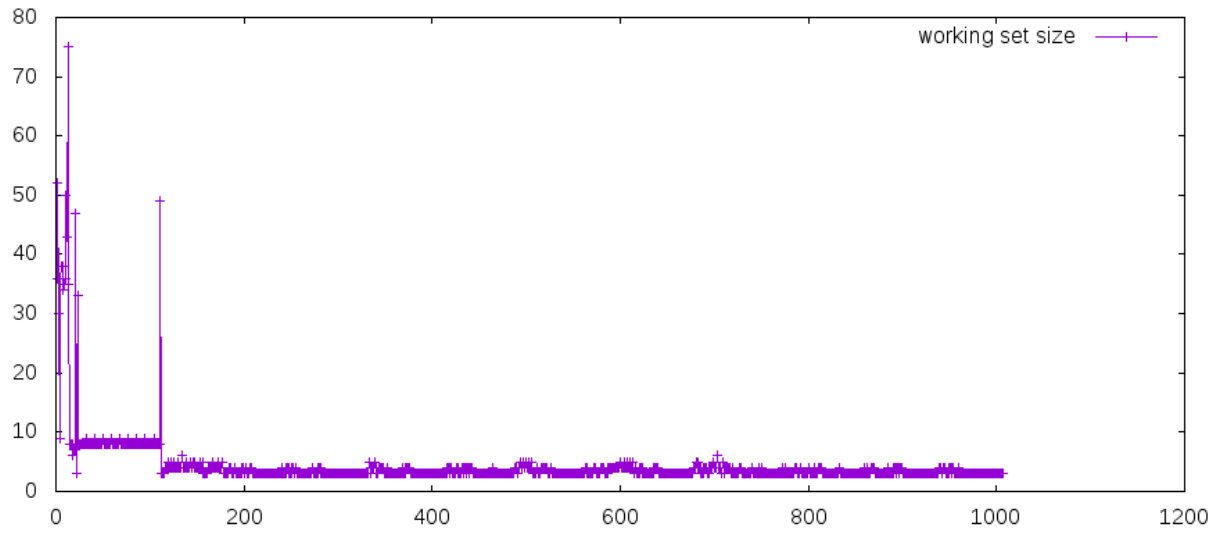
It is obvious that the working set size at each time instance is mostly determined by the program you are running, to be more specific, it is related to the algorithm of your program, that is how often you make memory references and how many bytes is related to the memory reference, and the reason of that is graphs for heapsort, quicksort and radixsort are all different with each other.

(b) Difference between input sizes and window size:

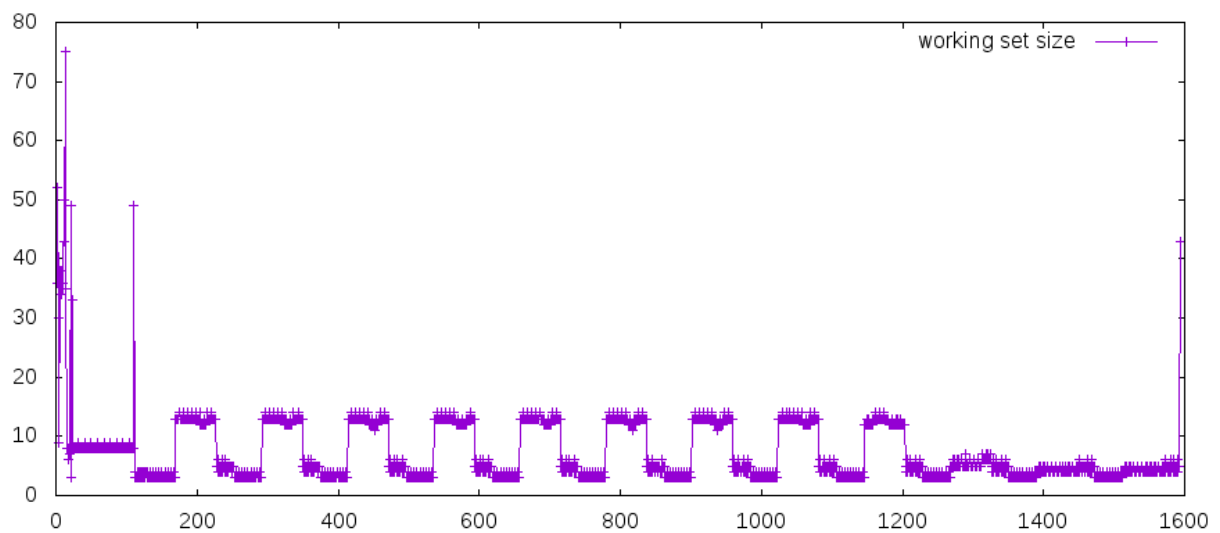
Heapsort 10000 elements, pagesize 4096, window size 10000

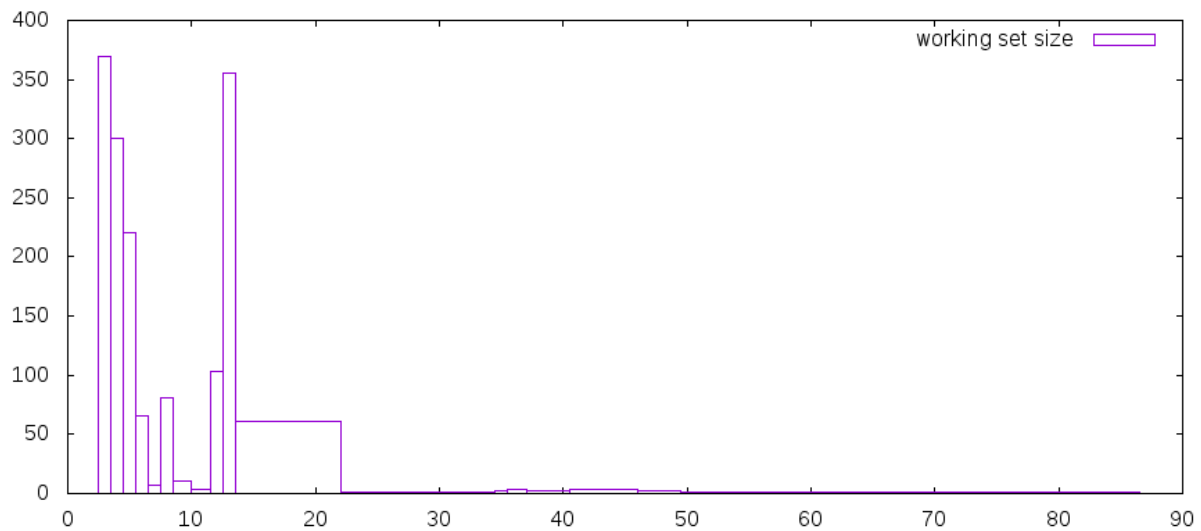


Quicksort 10000 elements, pagesize 4096, window size 10000



Radixsort 10000 elements, pagesize 4096, window size 10000

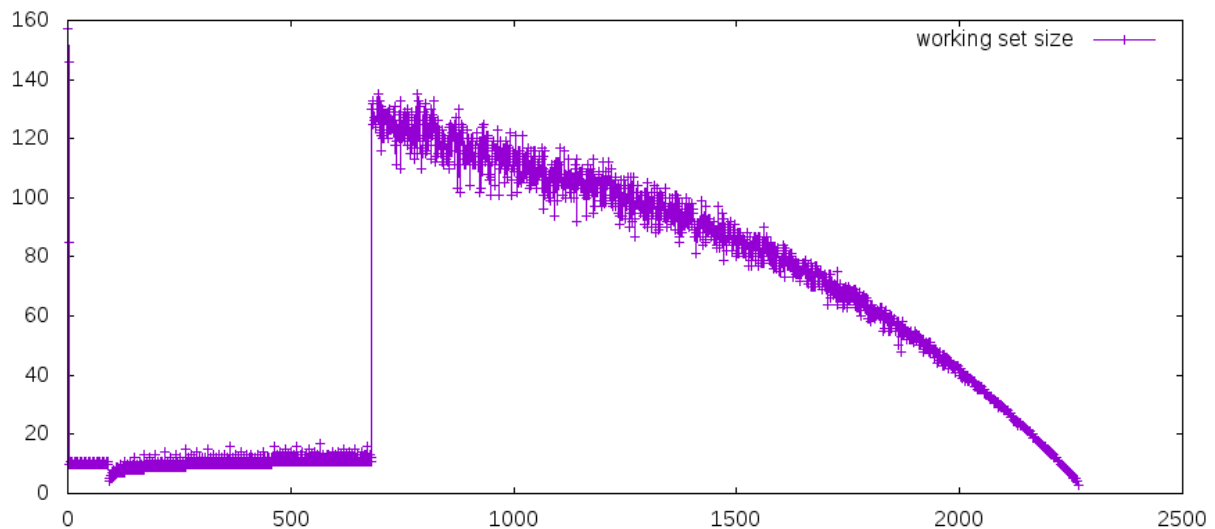


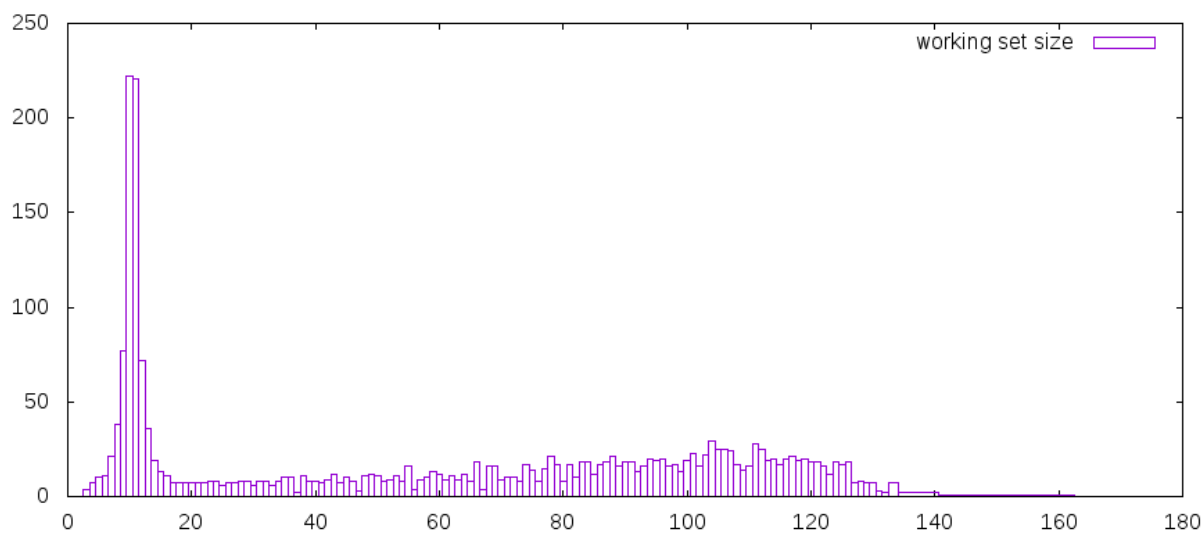


If the difference between window size and input size is too large, then the output will be general and vague, so I modify the window size and input size at the same time to do the analysis. As the diagrams show, the overall trend of the graph of three algorithms are not changing, and the only important difference is that the working set size at each time instance might be smaller.

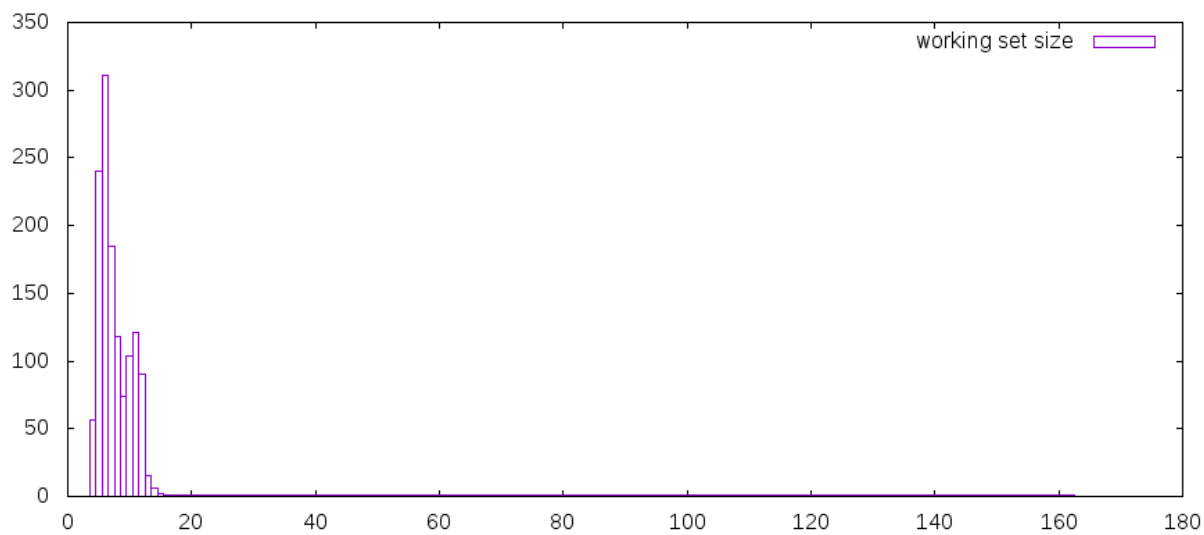
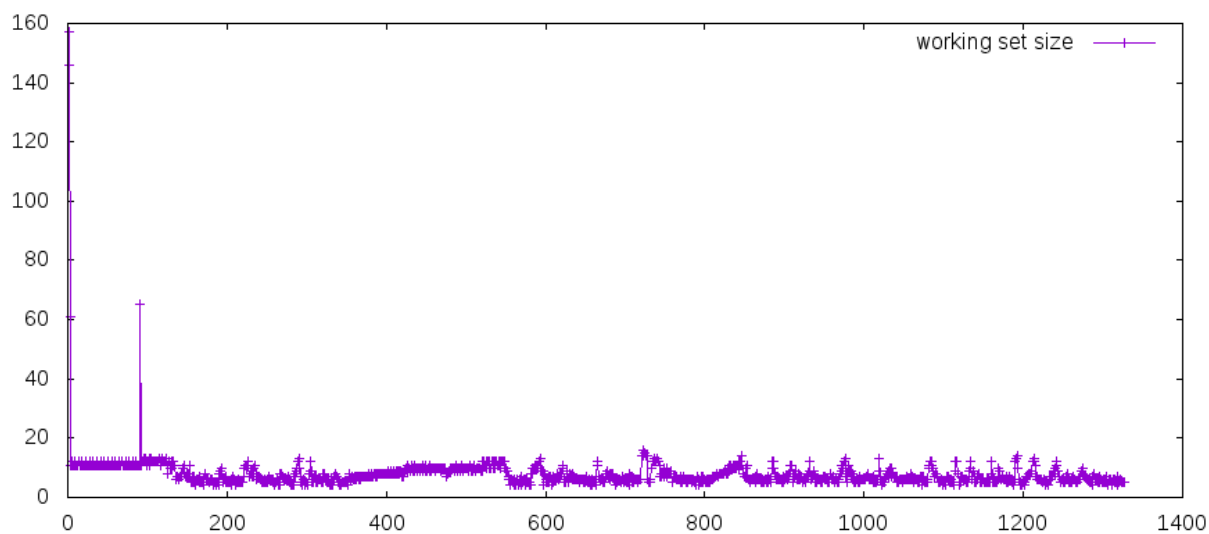
(c) Difference between pagesize

Heapsort 100000 elements, pagesize 2048, window size 100000

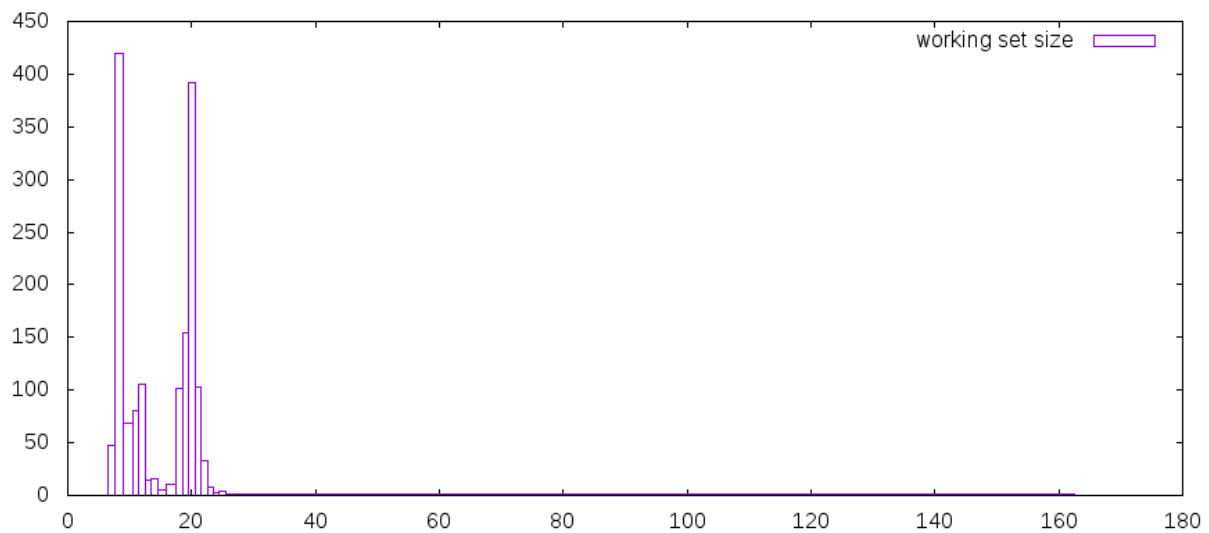
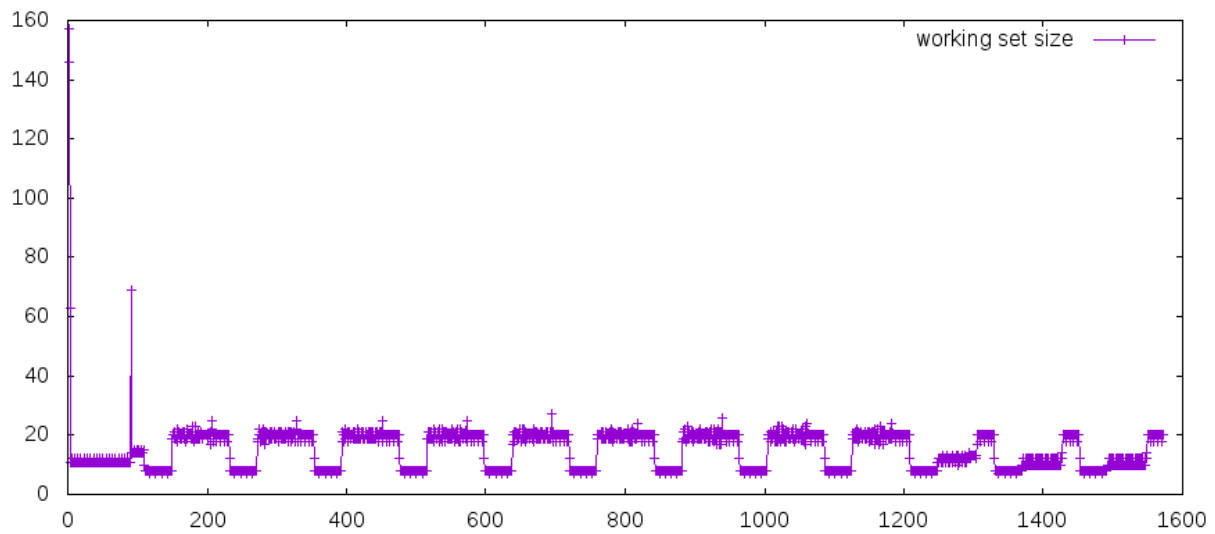




Quicksort 100000 elements, pagesize 2048, window size 100000



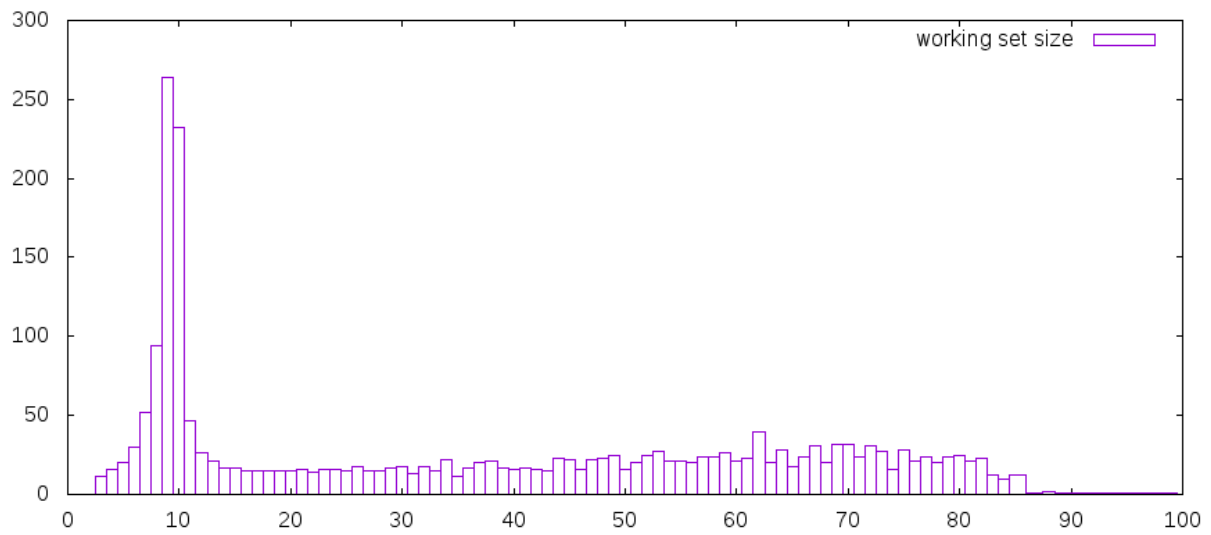
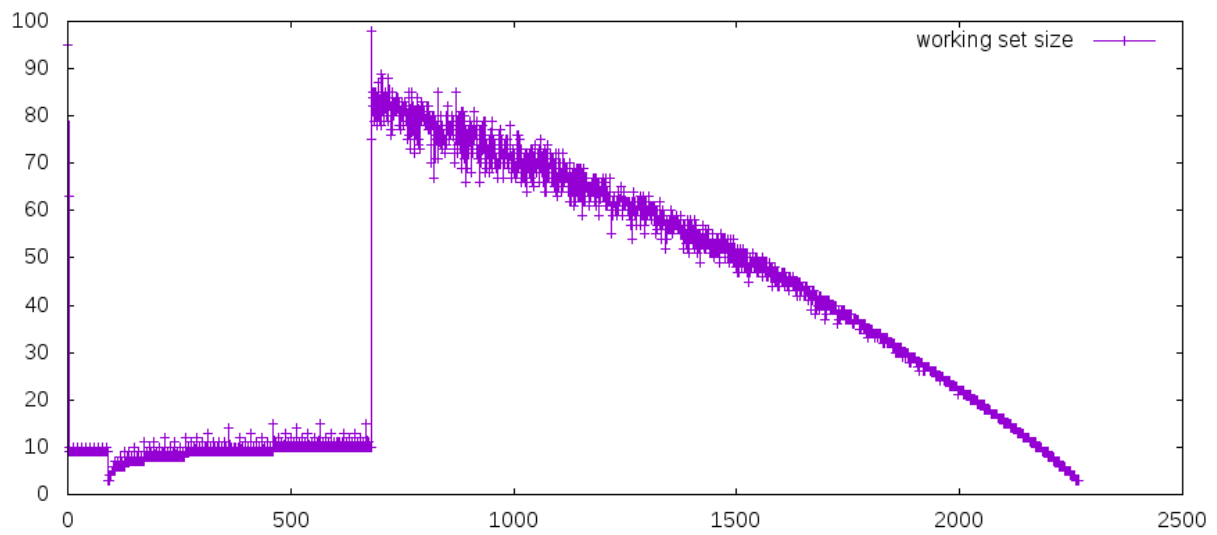
Radixsort 100000 elements, pagesize 2048, window size 100000



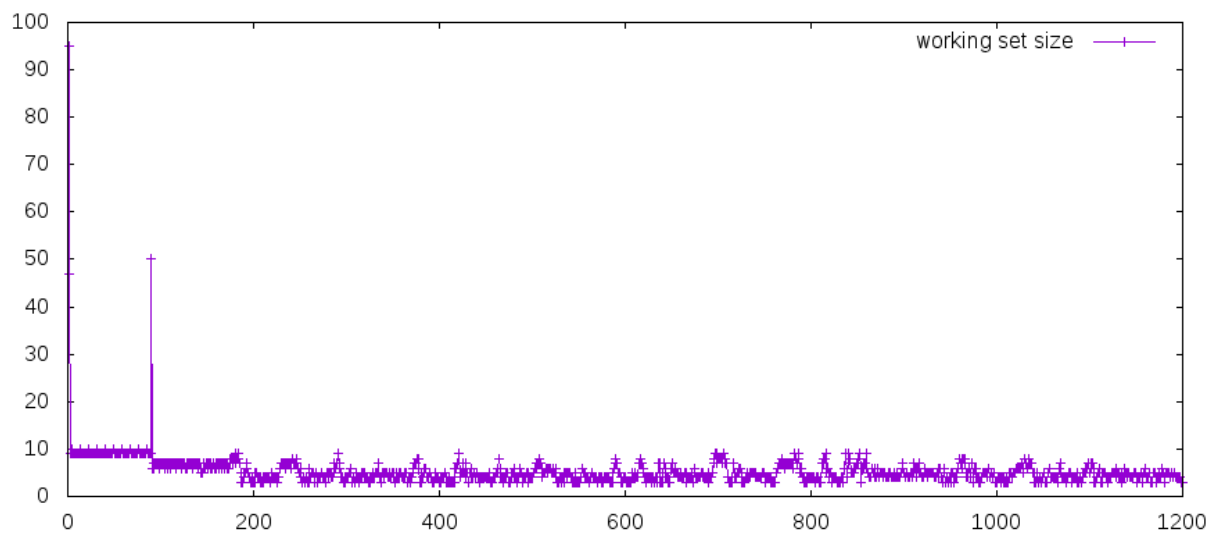
Again, it is obvious that change in pagesize will not influence the overall trend of diagram of all the three algorithms, however, the working set size at each time instance grows dramatically. The growth is reasonable since there is a decrease in the pagesize.

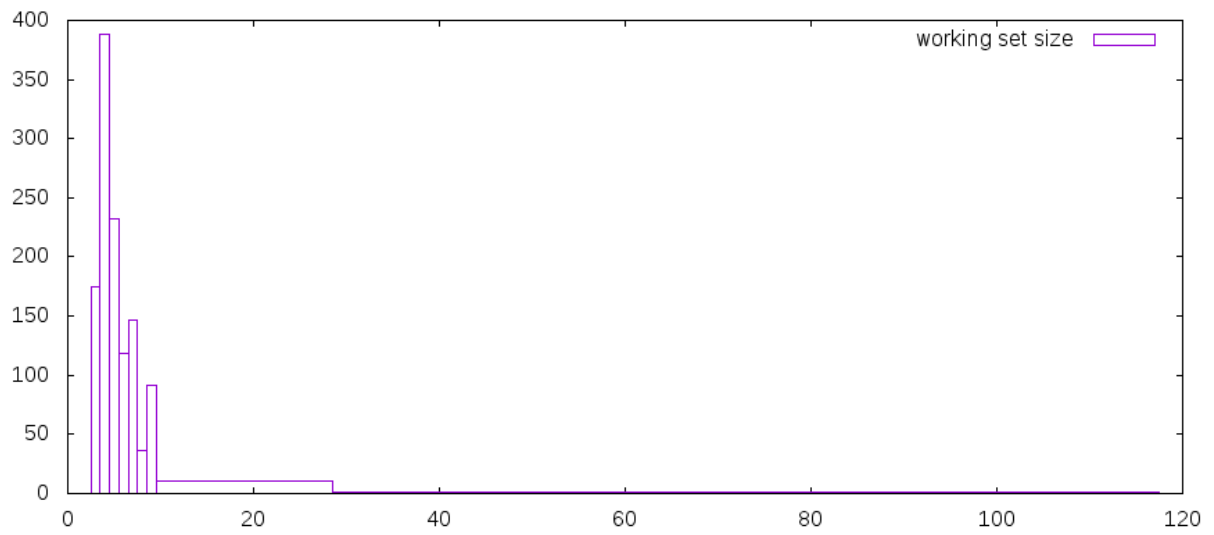
(d) Difference between skipsize

Heapsort 100000 elements, skipsize 100000, pagesize 4096, window size 100000

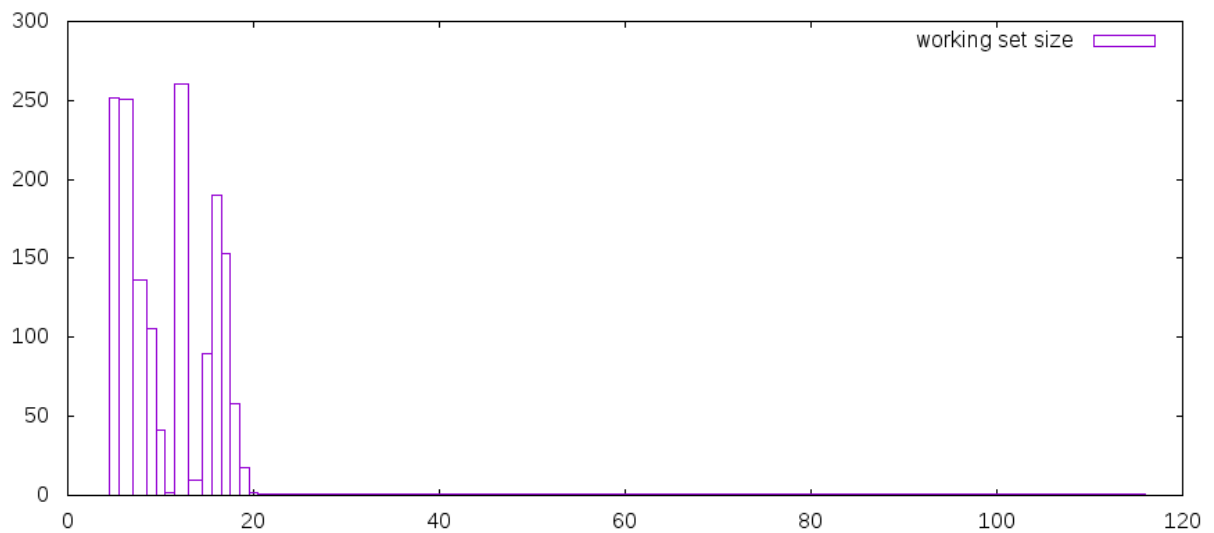
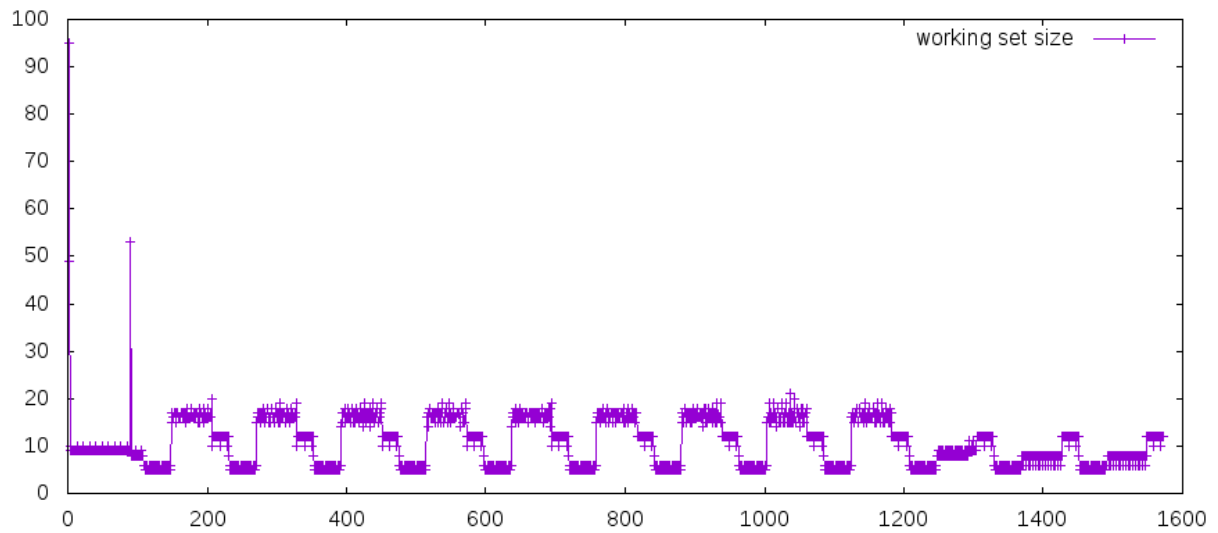


Quicksort 100000 elements, skipsize 100000, pagesize 4096, windowsize 100000





Radixsort 100000 elements, skipsize 100000, pagesize 4096, window size 100000



According to the graphs, considerable size of skipsize can effectively skip the beginning time period, which is the time period that the working set size is significantly larger than the rest of the time. This can be revealed by the graphs of all the three algorithms, the starting of each graph has less uncommonly large working set size.