On the time and space complexity of computation using write-once memory, or Is pen really much worse than pencil?


The authors study a RAM (random access machine) model with a small amount of regular memory (such as constant, logarithmic or $n^\alpha$ for $\alpha < 1$, where $n$ is the size of the input) and a polynomial amount of write-once memory. In this latter memory part bits may be set but not reset.

In converting any algorithm for an ordinary polynomial time-bounded RAM to one for this new model, the running time need only be increased by a factor of $O(\log n / \log \log n)$. For a few special cases—viz. for oblivious algorithms and for persistent computations—better upper bounds are established. (An algorithm is called oblivious if its read/write access pattern only depends on the input length. A computation is persistent if, at all times, the memory state of the computation at any previous moment can be reconstructed.) Any computation on this write-once model can be made persistent without a dramatic increase of time or space.

General simulations of an ordinary RAM by this write-once model require space proportional to the number of steps simulated. In investigating the space complexity the authors define an analogue of the pebbling game, called the pebble-sticker game. A sticker differs from a pebble in that it cannot be removed once it has been placed in the computation graph. The class of problems which can be solved with a polynomial amount of write-once memory happens to be equal to P, the class of problems solvable deterministically in polynomial time.