A lower bound for adding $n$ integers

**Problem** adding $n$ integers

**Input:** $x[0], \ldots, x[n-1]$ are integers (as large as $n$ bits), initially in memory cells $M[0], \ldots, M[n-1]$.

**Output:** $x[0] + \ldots + x[n-1]$ is left in $M[0]$.

**Algorithm** parAdd

**Theorem 14.13** Any Priority-Write PRAM with $p$ processors that computes parAdd must do at least $\lg(n) + 1 - \lg \lg(4p)$ steps.

**Corollary** Any CREW PRAM, Common-Write PRAM, Arbitrary-Write PRAM, or Priority-Write PRAM that computes parAdd must do at least $(\log n)$ steps if $p$ is bounded by any polynomial in $n$. 
Sketch of proof

- n integers, each of n bits, are stored in M[0], ..., M[n-1].
- Simplification (to reduce the size of input and output space)
  - Integer in M[i] is either 2 or 0
  - # of possible different inputs is 2^n and each input has a distinct sum.
- # of possible different outputs is 2^n.
- S_t: # of distinct states a processor can be in after t steps, with S_0 = 1.
- V_t: # of distinct values that could be in a memory cell after t steps, with V_0 = 2.
- p: # of processors.
- For t > 0, we have
  - S_t = S_{t-1} V_{t-1}
  - V_t = p S_t + V_{t-1}
- For t > 1, V_t = p S_t + V_{t-1} = pS_t + V_{t-1} = V_{t-1} (pS_{t-1} + 1) ≤ V_{t-1} (pS_{t-1} + V_{t-2}) = (V_{t-1})^2
- That is, V_t ≤ (V_{t-1})^2
- V_t ≤ (V_{t-1})^2 ≤ (V_{t-2})^2 ≤ ... ≤ (V_1)^2, where e = 2^{t-1}, and V_1 = 2p+2 ≤ 4p.
- That is, V_t ≤ (4p)^t
- In order to distinguish 2^n possible different outputs, a parallel algorithm has to have V_T ≥ 2^n after T steps.
- 2^n ≤ V_T ≤ (4p)^t, where d = 2^{T-1}
- Therefore, T ≥ Ω(n + 1 - lg(lg(4p))) = Ω(lg(n)) if p is bounded by a polynomial in n.