Communication costs of Schönhage-Strassen
fast integer multiplication

Derrick Coetzee, Jim Demmel, Oded Schwartz

Main results
- Tight lower and upper bounds on the communication cost of Schönhage-Strassen integer multiplication:
  \[ IO(n) = \Theta\left(\frac{n \log n}{\log M}\right) \]
  \(n\) is the length of the input numbers
  \(M\) is the fast memory size

Schönhage-Strassen
- FFT-based multiplication algorithm
- Multiplies two \(n\)-bit integers with \(O(n \log n \log \log n)\) arithmetic complexity
  - Compare long multiplication, \(O(n^2)\)
- Asymptotically fastest practical algorithm

The algorithm
Split inputs into \(d\)-length vectors of \(m\)-bit numbers (\(dm = n\))
  - Note: Can eliminate zero-padding using negacyclic convolution

- Compute acyclic convolution:
  - FFT, pointwise multiplication, IFFT
- Do all operations in ring \(\mathbb{Z}_{2^n+1}\)
- Multiplications during FFT become shifts/adds
- Modular reductions after recursive multiplications become shifts/adds

Arithmetic complexity
- Largest partial sum requires \(2m + \log d\) bits
  \(\Theta(n \log n)\) total work per level
  \(\Theta(\log \log n)\) levels
- \(T(n) = O(n \log n \log \log n)\)

Communication cost of FFT:
- \(IO = \Omega(n \log n / \log M)\)
  - [Hong & Kung 1981, Savage 1995]
- \(IO = O(n \log n / \log M)\)
  - [Frigo, Leiserson, Prokop, Ramachandran 1999]

Communication cost of Integer multiplication:
- **Phased** implementations:
  Each FFT done independently
  - \(IO(n) = \Theta\left(\frac{n \log n}{\log M}\right)\) if \(n > 3M\)
    - \(\Theta(M)\) otherwise
  - \(\Rightarrow IO(n) = \Theta\left(\frac{n \log n}{\log M}\right)\)
- **Interleaved** implementations cannot do better:
  - Proof: Impose reads/writes before/after each FFT
  - \(IO(n) \geq \Theta\left(\frac{n \log n}{\log M} - 2n\right)\) if \(n > 3M\)
    - \(\Theta(M)\) otherwise
  - \(c \frac{n \log n}{\log M} - 2n > c \frac{n \log n}{\log M}\) for \(n > M^{2/(c-c)}\)
  - Holds for all but \(O(1)\) recursion levels
  - \(\Theta(\log \log n)\) levels do \(O(n \log n / \log M)\) I/O each.

Future Work
- Apply to other multiplication algorithms:
  - [Strassen 1968],[Knuth 1997],[Fürer 2007],
    - [De, Saha, Kurur, Saptharishi 2008]
- Extend to other multiplication algorithms:
  - [Karatsuba 1962],[Toom 1963],[Cook 1966]
- Hybrids of the above
- Apply to polynomials multiplication
- Implement, predict, and test performance
- CA-Parallel algorithms