Another Randomized Algorithm

Freivalds’ Algorithm for Matrix Multiplication
Randomized Algorithms

• Quicksort makes effective use of random numbers, but is no faster than Mergesort or Heapsort.

• Here we will see a problem that has a simple randomized algorithm faster than any known deterministic solution.
Matrix Multiplication

Multiplying $n \times n$ matrices ($n = 2$ in this example)

$$
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
\begin{bmatrix}
w & x \\
y & z
\end{bmatrix}
=
\begin{bmatrix}
aw + by & ax + bz \\
cw + dy & cx + dz
\end{bmatrix}
$$

Complexity of straightforward algorithm: $\Theta(n^3)$ time

(There are 8 multiplications here; in general, $n$ multiplications for each of $n^2$ entries)

Coppersmith & Winograd showed how to do it in time $O(n^{2.376})$ in 1989.

Williams improved this to $O(n^{2.3729})$ in 2011. Progress!
History of Matrix Multiplication Algorithms
Running time: $O(n^\omega)$
Frievalds’ Algorithm (1977)

- Freivalds’ variant of problem:
  Determine whether $n \times n$ matrices $A$, $B$, and $C$ satisfy the condition $AB = C$

- Method:
  - Choose $x \in \{0,1\}^n$ randomly and uniformly (vector of length $n$)
  - If $ABx \neq Cx$ then report “$AB \neq C$”
    else report “$AB = C$ probably”
Running Time

- $ABx = A(Bx)$, so we have 3 instances of an $n \times n$ matrix times an $n$-vector
- These are $O(n^2)$ time operations if done straightforwardly
- Total running time $O(n^2)$
- Fastest deterministic solution known: $O(n^{2.3729})$
How Often Is It Wrong?

\[ P(ABx = Cx \mid AB = C) = 1 \]

\[ P(ABx = Cx \mid AB \neq C) \leq \frac{1}{2} : \]

- Assume \( AB \neq C \)
- Then \( AB - C \neq 0 \), so there exist \( i, j \) with \( (AB - C)_{ij} \neq 0 \)
- Let \( (d_1, d_2, \ldots, d_n) \) be \( i \)-th row of \( AB - C \); \( d_j \neq 0 \)
- \( P((AB - C)x = 0 \mid AB \neq C) \)
  \[ \leq P(\sum_{i=1}^{n} d_i x_i = 0 \mid AB \neq C) \]
  \[ = P(x_j = -\frac{1}{d_j} \sum_{i \neq j} d_i x_i \mid AB \neq C) \]
  \[ \leq \frac{1}{2} \]
Decreasing the Probability of Error

- By iterating with $k$ random, independent choices of $x$, we can decrease probability of error to $1/2^k$, using time $O(kn^2)$.

- Interesting comparison
  - Quicksort is always correct, and runs slowly with small probability.
  - Frievalds’ algorithm is always fast, and incorrect with small probability.