CS 681: Computational Number Theory and Algebra **Lecture 23**
Lecturer: Manindra Agrawal **Notes by: Ashwini Aroskar**

September 30, 2005.

Recall $|B| \geq (\ln n)^r$ where $r = \frac{\ln n}{\ln k}$

Time Complexity of Dixon's Algorithm

Expected number of iterations in Dixon's algorithm to get a single pair $(a, b) \leq (\ln n)^r$

Time complexity of the algorithm $= \tilde{O}(k^3 + k^2(\ln n)^r)$

The goal now is to choose k such that the time complexity is minimized.
The time complexity for matrix multiplication, using Gaussian elimination, is $\bigcirc(k^3)$, but this can be reduced for sparse matrices.

Sparse Matrix has l non-zero entries out n^2.
Theorem: There is an algorithm to invert such a matrix with time complexity $\bigcirc(nl)$.

The $(t+1)$ x t matrix (α_{ij}) has at most $(t+1)\log n$ non-zero entries.
So, the time complexity of finding a non-trivial vector in the null-space of this matrix $= \bigcirc(t^2 \ln n) = \bigcirc(k^2 \ln n)$

Improved time complexity of Dixon's Algorithm $= \tilde{O}(k^2(\ln n)^r)$

$k^2(\ln n)^r = e^{2\ln k + r \ln n \ln\ln n} = e^{2\ln k + \frac{\ln n \ln\ln n}{\ln k}}$
Minimum is achieved when $2\ln k = \frac{\ln n \ln\ln n}{\ln k}$, that is, when $\ln k = \frac{1}{\sqrt{2}}(\ln n \ln\ln n)^{\frac{1}{2}}$

Time complexity $= \tilde{O}(e^{2\sqrt{2}(\ln n \ln\ln n)^{\frac{1}{2}}})$

Quadratic Sieve

Let $c = \lfloor\sqrt{n}\rfloor$
Consider numbers of the form $(c+l)^2 - n$.
$(c+l)^2 - n \approx 2cl + l^2$
If $l \leq n^\epsilon$ then $(c+l)^2 - n = \bigcirc(n^{\frac{1}{2}+\epsilon})$.

We choose b among these numbers as the fraction of k-smooth numbers among them is higher. Although this is strongly implied by certain conjectures, there is no proof.