Last class: Polynomial identity testing

Schwartz-Zippel alg:

For polynomial $P(x_1, \ldots, x_n)$ of degree $\leq d$,
choose $r_1, \ldots, r_n$ u.a.r. from S,
then $\Pr(P(r_1, \ldots, r_n) = 0 \mid P \neq 0) \leq d/|S|$

Perfect matchings of bipartite graphs:

For $G = (L \cup R, E)$, let $n \times n$ matrix A be:

$$a_{ij} = \begin{cases} x_{ij} & \text{if } (i,j) \in E \\ 0 & \text{o/w} \end{cases}$$

where the x_{ij}'s are variables.

Then, $\det(A) \neq 0$ iff G contains a perfect matching.
(can generalize to non-bipartite graphs, see homework)

Moreover, if G contains a perfect matching then we can iteratively construct a perfect matching by checking if $\det(A_{ij}) \neq 0$ for an edge (i,j)

where $A_{ij} = A$ with row i & column j removed,

and recursing on $G \setminus \{i,j\}$ if it is $\neq 0$.

But can we find a perfect matching in parallel?

Can all edges simultaneously check if $\det(A_{ij}) \neq 0$?

If there is a unique perfect matching ~~we~~ that all edges are checking against then we can do it in parallel.

<u>Isolation Lemma</u>: [Mulmuley, Vazirani, Vazirani '87]

Let $S_1, \ldots, S_k$ be subsets of a set S where $|S| = m$.

For each $x \in S$, choose $w(x)$ u.a.r. from $\{1, \ldots, l\}$.

Then, $\Pr\left(\exists \text{ unique subset } S_i \text{ of min weight}\right) \geq 1 - \frac{m}{l}$

where $w(S_i) = \sum_{x \in S_i} w(x)$.

Now let's apply the isolation lemma to the perfect matching problem.

For each edge (i,j), choose its weight w_{ij} u.a.r. from $\{1, \ldots, 2m\}$.

Hence, we know that with prob. $\geq 1/2$ there is a unique perfect matching of min weight. But this weights a matching M as $w(M) = \sum_{e \in M} w(e)$.

And in the $\det(A)$ the weight is $\prod_{e \in M} x_e$

Thus, in matrix A,

replace x_{ij} by $2^{w_{ij}}$

Denote this matrix as D.

We know that if G does not contain a perfect matching then $\det(D) = 0$ since $\det(A) = 0$.

What if G has a perfect matching?

<u>Lemma:</u> If there is a unique perfect matching of min weight, then $\det(D) \neq 0$ & the max power of 2 dividing $\det(D)$ is $W^* =$ min weight of perfect matching

$$= \min_{M \in \mathcal{P}} \sum_{e \in M} w(e)$$

where $\mathcal{P}$ = set of all perfect matchings in G.

Proof: For $\mathcal{P}$ = set of perfect matchings, we're assuming $\mathcal{P} \neq \emptyset$ & that there is a unique perfect matching of min weight, denote it as M^*, and $W^* = w(M^*)$.

$$\det(D) = \sum_{M \in \mathcal{P}} (-1)^{\mathrm{sgn}(M)} \prod_i 2^{w_{i m(i)}} = \sum_{M \in \mathcal{P}} (-1)^{\mathrm{sgn}(M)} 2^{\sum w_{i m(i)}}$$

$$= \sum_{M \in \mathcal{P}} \pm 2^{w(M)} = \pm 2^{W^*} + \sum_{j > W^*} k_j 2^j \quad \text{for some integers } k_j$$

Since exactly one $M^* \in \mathcal{P}$ with $w(M^*) = W^*$

all $M' \neq M^*$ have $w(M') > W^*$

∎

We now have a parallel alg:

1. For each $(i,j) \in E$, pick w_{ij} u.a.r. from $\{1, \dots, 2m\}$
2. Compute $\det(D)$ (this can be done in parallel)
3. If $\det(D) = 0$ then output NO perfect matching
4. Let W^* be the max i where 2^i divides $\det(D)$.
5. For each $(i,j) \in E$:
   a. Evaluate $\det(D_{ij})$
   b. If $\det(D_{ij}) = 0$ then stop considering (i,j).
   c. Else, find max j s.t. 2^j divides $\det(D_{ij})$ and denote it as W^*_{ij}.
   d. If $W^*_{ij} + w_{ij} = W^*$ then output edge (i,j)
6. Finally, check that the outputted edges form a perfect matching.

Note, $\Pr(\text{alg. outputs a perfect matching}) \geq 1/2$

Alice has a n-bit number $a = a_0 a_1 \ldots a_{n-1}$

Bob has a n-bit number $b = b_0 b_1 \ldots b_{n-1}$

where n is HUGE.

Can they quickly check if $a \stackrel{?}{=} b$

Fingerprinting:

Alice picks a prime number p u.a.r. from $\{2, \ldots, T\}$
where T will be specified later.

She computes $F_p(a) = a \bmod p$.

She sends p & $F_p(a)$ to Bob.

Bob computes $F_p(b) = b \bmod p$

& checks if $F_p(a) = F_p(b)$?

If $a = b$ then we always have $F_p(a) = F_p(b)$.

But if $a \neq b$ then we might still have $F_p(a) \stackrel{?}{=} F_p(b)$.

For an integer $x>0$, let $\pi(x) = \#$ of primes $\leq x$.

Prime number theorem: $\lim_{x\to\infty} \pi(x) = \frac{x}{\ln x}$

Moreover, for $x \geq 17$, $\frac{x}{\ln x} \leq \pi(x) \leq \frac{1.26x}{\ln x}$

Back to the original problem,

If $F_p(a) = F_p(b)$ then $a \equiv b \mod p$

so $a-b \equiv 0 \mod p$

which means p divides $|a-b|$

Note, $|a-b|$ is $\leq n$ bits,

and thus $\leq n$ primes divide $|a-b|$

(since each prime is ≥ 2)

actually $\leq \pi(n)$ primes divide $|a-b|$,

(that will save a log factor)

Therefore,

$$\Pr\left(F_p(a)=F_p(b) \mid a\neq b\right) \leq \frac{\pi(n)}{\pi(T)}$$

$$\leq 1.26\frac{n}{\ln n}\frac{\ln T}{T}$$

let $T = cn$,

$$\leq \frac{1.26}{c}\frac{\ln(cn)}{\ln n} = \frac{1.26}{c}\left(1+\frac{\ln c}{\ln n}\right)$$

this is small even for $c=10$,
and then can run t-trials to
reduce the error prob exponentially in t.

Note, since $T = O(n)$,
then p ($\& F_p(a)$) are $O(\lg n)$ bits.

Applications of fingerprinting:

- Polynomial identity testing: intermediate computations when evaluating polynomials at $(r_1, \ldots, r_n)$ may be HUGE numbers. Can do modulo a small prime as in the fingerprinting scheme.

- Pattern matching:

Binary text $X = x_1 \ldots x_n$ for large n

& shorter text $Y = y_1 \ldots y_m$ for $m \ll n$.

Does Y occur as a contiguous substring of X?

I.e., for j, let $X(j) = x_j x_{j+1} \ldots x_{j+m-1}$

Does there exist a j where $X(j) = Y$?

Naive alg.: $O(mn)$ time.

Sophisticated algorithms in $O(m+n)$ time due to [Boyer Moore 77] & [Knuth Morris Pratt 77].

Here's a simple $O(m+n)$ time alg. due to [Karp, Rabin '81].

Pick random prime $p \in [2, \ldots, T]$

Compute $F_p(Y) = Y \bmod p$

for $j = 1 \rightarrow$ ~~n-m~~ $n-m+1$:

- Compute $F_p(X(j))$
- If $F_p(Y) = F_p(X(j))$ then output match@j & halt.

Output No match.

$$\Pr\left(F_p(Y) = F_p(X(j)) \mid Y \neq X(j)\right) \leq \pi(m) / \pi(T)$$

thus: $$\Pr\left(\text{output match} \mid \text{No match}\right) \leq \frac{n\pi(m)}{\pi(T)}$$

But, note that if p divides $|Y - X(j)|$ for some j

then p divides $\prod_j |Y - X(j)|$

and this is $\leq nm$ bits long.

Hence, we have a better bound: (replace $n\pi(m)$ by $\pi(nm)$)

$$\Pr(\text{error}) \leq \frac{\pi(nm)}{\pi(T)}$$ which is small for $T = 10mn$.

Running time:

Note, p has $O(\log(mn)) = O(\log n)$ bits,

So let's assume arithmetic mod p in $O(1)$ time.

Since Y is m bits, then

Computing $F_p(Y)$ takes $O(m)$ time

Need to compute $F_p(X(j))$ for all j:

Naive: $O(m)$ time each & thus $O(nm)$ total time.

But $X(j)$ & $X(j+1)$ are similar,

$$X(j+1) = 2\left(X(j) - 2^{m-1}x_j\right) + x_{j+m}$$

Thus,

$$F_p(X(j+1)) = \left(2\left(F_p(X(j)) - 2^{m-1}x_j\right) + x_{j+m}\right) \bmod p$$

which takes $O(1)$ time since this is either 0 or 2^{m-1} which can be precomputed & rest are $O(1)$ arithmetic ops.

$\Rightarrow O(n+m)$ total time.