Alg. Comb.: Lesson Plan 2 — 28/1/2015

I. Matrix multiplication

A. Motivation

1. Ultimate goal: randomised algorithm for perfect matchings.
2. Warm-up: matrix multiplication verification
   a. Exposure to rand. alg. ideas we will use later
   b. M.M. of fundamental importance in TCS.

B. Complexity

1. Natural algorithm: $O(n^3)$ operations
   a. Remark: parallelisable → $O(n)$ time × $O(n^2)$ processors
2. $\exists$ very clever algorithms that are more efficient
   a. Current record: $O(n^{2.376})$
      (i) Theoretical interest only → hidden constants
   b. c.f. Homework.

C. Verification

1. Our friend Gauss has a very fast, very secret algorithm for multiplying matrices
   a. Joke: employs child labour
   b. We give him two non matrices A and B, he returns C, and claims $C = AB$.
2. Trust, but verify: would like to confirm the correctness of his output
   a. Could compute $A \times B$ and compare, but then we lose all the efficiency of using Gauss in the first place.
   b. ∴ want quick but reliable way to verify product.
      (i) ⇒ randomised algorithms

D. Naive randomised algorithm

1. Randomly select ℓ entries to check
   (a) Compute correct values for those ℓ
   (b) If mismatch → Reject!
   (c) Else → Accept.
2. Efficiency: takes $O(\ell n)$ operations
3. Accuracy: suppose one entry is incorrect
   (i) Probability of finding $= \frac{\ell}{n^2}$
4. e.g.: $\ell = n \Rightarrow O(n^2)$ time but only $O(\frac{1}{n})$ prob. of finding an error.

E. Clever randomised algorithm

1. Zero testing
   a. More general problem: test if a matrix D is zero.
   b. No access to entries of D, but 'oracle' access: can multiply vectors.
   c. In our situation, $D = C - AB$
      (i) Oracle query requires $O(n^2)$ operations.
2. Random vector
   a. Choose $\underline{x} \in \{0,1\}^n$ uniformly at random
   b. Claim: If $D \equiv 0$, then $D\underline{x} = \underline{0}$ ✓
   c. Claim: If $D \neq 0$, then $\mathbb{P}(D\underline{x} = \underline{0}) \leq \frac{1}{2}$.
      Pf: Suppose $D_{i_0 j_0} \neq 0$. Then
      $$(D\underline{x})_{i_0} = \sum_j D_{i_0 j} x_j = D_{i_0 j_0} x_{j_0} + \sum_{j \neq j_0} D_{i_0 j} x_j$$
      Fix x_j, $j \neq j_0$. Then at most one of $x_{j_0} \in \{0,1\}$ can make this equal to zero. □
   d. ∴ $\mathbb{P}(\text{false accept}) \leq \frac{1}{2}$.
      $\mathbb{P}(\text{false reject}) = 0$.
3. Repeated trials:
   a. k repeats: $O(kn)$ time, $\mathbb{P}(\text{false accept}) \leq 2^{-k}$.

(Remark. H/W: deterministic $O(n^2)$ verification.)

II. Zero Polynomial Testing

A. Connection to matrix multiplication verification

1. $n \times n$ matrix D = linear function in n variables
   $$D: \mathbb{F}^n \to \mathbb{F}^n \qquad (\mathbb{F} \text{ some field})$$
   $$\underline{x} \mapsto D\underline{x}$$
2. to test if an arbitrary fn is identically zero: need to test vs every possible input.
3. Linear maps are sufficiently rigid $\Rightarrow$ allows for efficient testing.
4. Polynomials also share similar rigidity.
5. Given an n-variable polynomial $p \in \mathbb{F}[x_1, \ldots, x_n]$, determine if $p \equiv 0$.
   (i) As before, may not have direct access to coefficients $\to$ oracle evaluation $\underline{x} \mapsto p(\underline{x})$.
   (ii) Even with coefficients, not obvious; eg. $x^2 + x$ in $\mathbb{F}_2[x]$.

B. Univariate case

1. Take $n = 1$, $p: \mathbb{F} \to \mathbb{F}$ polynomial of degree d
2. (Ask class for ideas) Fundamental Theorem of Algebra
   $\Rightarrow p \equiv 0$ or at most d zeroes
3. Test p against (any) $d+1$ points
   a. If any non-zero $\longrightarrow$ NO
   b. If all zero $\longrightarrow$ YES
4. Deterministic, exact algorithm.
5. Class exercise: what if $|\mathbb{F}| \leq d$?

C. Multivariate case

1. Definition
   a. $p \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ is a finite sum of monomials: $a_{d_1, \ldots, d_n} x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$.

b. degree of the monomial = $d_1 + d_2 + \cdots + d_n$

c. degree of the polynomial = max degree of a (non-zero) monomial.

2. Complicated zero-sets

a. $p(x_1, x_2) = x_1^2 + x_2^2 - 1$, $\mathbb{F} = \mathbb{R}$

(i) degree = 2

(ii) zero-set = unit circle 

b. $p(x_1, x_2) = x_1 x_2 - 1$, $\mathbb{F} = \mathbb{R}$

(i) degree = 2

(ii) zero-set = hyperbola 

3. Schwartz-Zippel

a. Idea: cannot have too many zeroes inside a cube.

b. Thm 1: (Schwartz-Zippel)

Let $p(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero polynomial of degree $d \geq 0$, and $S \subseteq \mathbb{F}$ a finite set. Then $\#\{(r_1, \ldots, r_n) \in S^n : p(r_1, \ldots, r_n) = 0\} \leq d|S|^{n-1}$.

c. Proof: Induction on n.

(i) Base case: $n = 1 \rightsquigarrow$ F.T.A.

~~(ii) Induction step. Let $Z = \{(r_1, \ldots, r_n) \in S^n : p(r_1, \ldots, r_n) = 0\}$~~

~~$Z = Z_1 \sqcup Z_2$, where~~

~~$Z_1 = \{(r_1, \ldots, r_n) \in Z :$~~

(ii) WLOG x_1 appears in p, with max degree k, say.

$\Rightarrow p(x_1, \ldots, x_n) = \sum_{i=0}^{k} x_1^i p_i(x_2, \ldots, x_n)$.

(iii) Let $Z = \{(r_1, \ldots, r_n) \in S^n : p(r_1, \ldots, r_n) = 0\} = Z_1 \sqcup Z_2$,

where $Z_1 = \{(r_1, \ldots, r_n) \in Z : p_k(r_2, \ldots, r_n) = 0\}$

and $Z_2 = \{(r_1, \ldots, r_n) \in Z : p_k(r_2, \ldots, r_n) \neq 0\}$

(iv) By induction, since $\deg(p_k) \leq d - k$,

$\leq (d-k)|S|^{n-2}$ $(r_2, \ldots, r_n)$ s.t. $p_k = 0$

$\Rightarrow |Z_1| \leq (d-k)|S|^{n-1}$.

(v) $|S|^{n-1}$ choices for $r_2, \ldots, r_n$ in $Z_2 \rightsquigarrow$ deg k poly in x_1

FTA $\Rightarrow k|S|^{n-1} \geq |Z_2|$ ∎

4. Randomised algorithm
   a. Fix some $S \subseteq \mathbb{F}$, $|S| = 2d$
   b. Choose some $\underset{\sim}{x} \in S^n$ unif. at. random
   c. If $p(\underset{\sim}{x}) = 0 \rightsquigarrow$ Yes
      $p(\underset{\sim}{x}) \neq 0 \rightsquigarrow$ No
   d. $\mathbb{P}(\text{False No}) = 0$
   e. By Schwartz-Zippel, $\mathbb{P}(\text{False Yes}) \leq \frac{d|S|^{n-1}}{|S|^n} = \frac{1}{2}$.

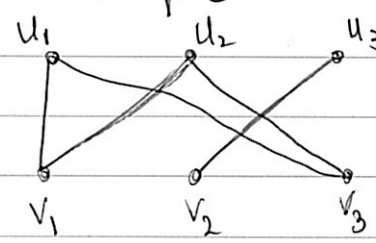
III. Testing for Perfect Bipartite Matchings

A. Framework

1. Bipartite graph $G = (U \cup V, E)$, $U = \{u_1, \dots, u_n\}$ and $V = \{v_1, \dots, v_n\}$.
2. Bipartite adjacency matrix $B \in \mathbb{F}^{\{0,1\}^{n \times n}}$
$$b_{ij} = \begin{cases} 1 & \text{if } \{u_i, v_j\} \in E \\ 0 & \text{o/w.} \end{cases}$$
3. Perfect matching $\iff \sigma \in S_n : \{\{u_1, v_{\sigma(1)}\}, \dots, \{u_n, v_{\sigma(n)}\}\} = M$.

B. Permanent & Determinant

1. Permutation π gives matching iff $b_{1,\pi(1)} b_{2,\pi(2)} \cdots b_{n,\pi(n)} = 1$.
2. $\therefore$ # perfect matchings $= \sum_{\pi \in S_n} b_{1,\pi(1)} b_{2,\pi(2)} \cdots b_{n,\pi(n)} =: \text{per}(B)$
   a. NP-hard to compute!
3. determinant = signed version of the permanent
$$\det(B) = \sum_{\pi \in S_n} \text{sgn}(\pi)\, b_{1,\pi(1)} b_{2,\pi(2)} \cdots b_{n,\pi(n)}.$$
   a. Can be efficiently computed.
   b. Can have cancellation.
4. Example



$$B = \begin{pmatrix} \boxed{1} & 0 & \textcircled{1} \\ \textcircled{1} & 0 & \boxed{1} \\ 0 & \textcircled{1} & 0 \end{pmatrix} \qquad \begin{aligned} \text{per}(B) &= 2 \\ \det(B) &= \textcircled{1} - \boxed{1} = 0 \end{aligned}$$

5. $\therefore \det(B) \neq 0 \Rightarrow \exists$ perf. matching
   $\not\Leftarrow$