Counting Problems

Two big pieces:

1. Equivalence of counting and generating via **self reducibility**
2. Generating via Markov chains

Volume

Outline:

- Describe problem. Membership oracle
- $\sharp P$ hard to volume intersection of half spaces in n dimensions
- In low dimensions, integral.
- even for convex bodies, can't do better than $(n/\log n))^n$ ratio
- what about FPRAS?

Estimating π:

- pick random in unit square
- check if in circle
- gives ratio of square to circle
- Extends to arbitrary shape with "membership oracle"
- Problem: rare events.
- Circle has good easy outer box

Problem: rare events:

- In 2d, long skinny shapes
- In high d, even round shape has exponentially larger bounding box

Solution: "creep up" on volume

- modify P to contain unit sphere B_1, contined in larger B_2 of radius r with r/r_1 polynomial
- choose $\rho = 1 - 1/n$.
- Consider sequence of bodies $\rho^i r P \cap B_2$
- note for large i, get P

- but for $i = 0$, body contains B_2
- so volume known
- so just need ratios
- At each step, need to random sample from $\rho^i rP \cap B_2$

Sample method: random walk forbidden to leave

- MC irreducible since body connected
- ensure aperiodic by staying put with prob. 1/2
- markov chain is "regular graph" so uniform stationary distribution
- eigenvalues show rapid mixing: after t steps, r.p.d at most

$$(1 - \frac{1}{10^17n^19})^t$$

- eigenvalues small because body convex: no bottlenecks.

Observations:

- Key idea of self reducibility: compare size of sequence of "related" shapes, then telescope ratios.
- Sizes compared by sampling
- Sample by markov chain
- wait: markov chain not exact?
- doesn't matter: just get accurate to within $(1 - 1/poly)$ in each step, product of errors still tiny.

Application: Permanent

Counting perfect matchings

- Choose random n-edge set
- check if matching
- problem: rare event
- to solve, need sample space where matchings are dense

Idea: self reducibility by adding an edge (till reach complete graph)

- problem: don't know how to generate random matching

Different idea: ratio of k-edge to $k-1$-edge matchings

- telescope down to 1-edge matchings (self reduction)
- in **dense graphs** (degree $n/2$), ratio is at most m^3.
- map each k edge matching by removing an edge: n^2 to 1
- map each $k-1$ edge matching to k-edge matching by **augmenting path** of length at most 3.
  - take unmatched u and v
  - if unmatched neighbor of u or v, done
  - by u and v have $n/2$ neighbors, so if all matched, some neighbor b of u matched to some neighbor a of v.
  - so each size k matching "receives" at most m^3 size $k-1$ matchings.

Generate via random walk

- based on using uniform generation to do sampling.
- applies to minimum degree $n/2$
- Let M_k be k-edge matchings, $\|M_k\| = m_k$
- algorithm estimates all ratios m_k/m_{k-1}, multiplies
- claim: ratio m_{k+1}/m_k polynomially bounded (dense).
- deduce sufficient to generate randomly from $M_k \cup M_{k-1}$, test frequency of m_k
- do so by random walk of local moves:
  - with probability 1/2. stay still
  - else Pick random edge e
  - if in M_k and e matched, remove
  - if in M_{k-1} end e can be added, add.
  - if in M_k, $e = (u, v)$, u matched to w and v unmatched, then match u to w.
  - else do nothing
  - Note that exactly one applies
- Matrix is symmetric (undirected), so double stochastic, so stationary distribution is uniform as desired.
- In text, prove $\lambda_2 = 1 - 1/n^{O(1)}$ on an n vertex graph (by proving expansion property)
- so within $n^{O(1)}$ steps, rpd is polynomially small
- so can pretend stationary

Recently, extended to non-dense case.

Coupling:

Method

- Run two copies of Markov chain X_t, Y_t
- Each considered in isolation is a copy of MC (that is, both have MC distribution)
- **but** they are not independent: they make dependent choices at each step
- in fact, after a while they are almost certainly the **same**
- Start Y_t in stationary distribution, X_t anywhere
- Coupling argument:

$$\begin{aligned}\Pr[X_t = j] &= \Pr[X_t = j \mid X_t = Y_t] \Pr[X_t = Y_t] + \Pr[X_t = j \mid X_t \neq Y_t] \Pr[X_t \neq Y_t] \\ &= \Pr[Y_t = j] \Pr[X_t = Y_t] + \epsilon \Pr[X_t = j \mid X_t \neq Y_t]\end{aligned}$$

  So just need to make ϵ (which is r.p.d.) small enough.

n-bit Hypercube walk: at each step, flip random bit to random value

- At step t, pick a random bit b, random value v
- both chains set but b to value v
- after $O(n \log n)$ steps, probably all bits matched.

Counting k colorings when $k > 2\Delta + 1$

- The reduction from (approximate) uniform generation
  - compute ratio of coloring of G to coloring of $G - e$
  - Recurse counting $G - e$ colorings
  - Base case k^n colorings of empty graph
- Bounding the ratio:
  - note $G - e$ colorings outnumber G colorings
  - By how much? Let L colorings in difference (u and v same color)
  - to make an L coloring a G coloring, change u to one of $k - \Delta = \Delta + 1$ legal colors
  - Each G-coloring arises at most one way from this
  - So each L coloring has at least $\Delta + 1$ neighbors unique to them
  - So L is $1/(\Delta + 1)$ fraction of G.
  - So can estimate ratio with few samples
- The chain:

- Pick random vertex, random color, try to recolor
- loops, so aperiodic
- Chain is time-reversible, so uniform distribution.

- Coupling:
  - choose random vertex v (same for both)
  - based on X_t and Y_t, choose bijection of colors
  - choose random color c
  - apply c to v in X_t (if can), $g(c)$ to v in Y_t (if can).
  - What bijection?
    * Let A be vertices that agree in color, D that disagree.
    * if $v \in D$, let g be identity
    * if $v \in A$, let N be neighbors of v
    * let C_X be colors that N has in X but not Y (X can't use them at v)
    * let C_Y similar, wlog larger than C_X
    * g should swap each C_X with some C_Y, leave other colors fixed. **Result:** if X doesn't change, Y doesn't

- Convergence:
  - Let $d'(v)$ be number of neighbors of v in opposite set, so

$$\sum_{v \in A} d'(v) = \sum_{v \in D} d'(v) = m'$$

  - Let $\delta = |D|$
  - Note at each step, δ changes by $0, \pm 1$
  - When does it increase?
    * v must be in A, but move to D
    * happens if only one MC accepts new color
    * If c not in C_X or C_Y, then $g(c) = c$ and both change
    * If $c \in C_X$, then $g(c) \in C_Y$ so neither moves
    * So must have $c \in C_Y$
    * But $|C_Y| \leq d'(v)$, so probability this happens is

$$\sum_{v \in A} \frac{1}{n} \cdot \frac{d'(v)}{k} = \frac{m'}{kn}$$

  - When does it decrease?
    * must have $v \in D$, only one moves

* sufficient that pick color not in either neighborhood of v,
* total neighborhood size 2Δ, but that counts the $d'(v)$ elements of A twice.
* so Prob.

$$\sum_{v \in D} \frac{1}{n} \cdot \frac{k - (2\Delta - d'(v))}{k} = \frac{k - 2\Delta}{kn}\delta + \frac{m'}{kn}$$

– Deduce that expected *change* in δ is difference of above, namely

$$-\frac{k - 2\Delta}{kn}\delta = -a\delta.$$

– So after t steps, $E[\delta_t] \leq (1-a)^t \delta_0 \leq (1-a)^t n$.

– Thus, probability $\delta > 0$ at most $(1-a)^t n$.

– But now note $a > 1/n^2$, so $n^2 \log n$ steps reduce to one over polynomial chance.

Note: couple depends on state, but who cares

- From worm's eye view, each chain is random walk
- so, all arguments hold

Counting vs. generating:

- we showed that by generating, can count
- by counting, can generate: