CSC 373 Lecture 31

Announcements:

Next assignment due Friday, Dec 2; term test 3 on Monday, Dec 5. last class Wed, Dec 7.

Today

- Go over question 2 of assignment
- Finish randomized algorithm for 2SAT and sketch extension to k-SAT
- Start compositeness (primality) testing

Random walk algorithm for 2-SAT

- It is not difficult to show that 2-SAT (determining if a 2CNF formula is satisfiable) is efficiently computable whereas we know that 3SAT is NP complete. We will provide a conceptually simple randomized algorithm to show that 2SAT is computationally easy. The same basic approach can be used to derive a randomized (which in turn leads to a deterministic) algorithm for 3SAT that runs in time [poly(n) *(4/3)^n*]. It is a big open question if one can get time *2^{o(n)}* algorithm for 3-SAT. The best known randomized time bound for 3-SAT is around (1.324)^n.

Stationary distribution

- Fact: If G is not bipartite, a stationary distribution exists for a uniform random walk on graph and that distribution is the vector

  $pi = <(d_1)/2m, ...,(d_n)/2m)>$ where d_i = degree of v_i and $m = |E|$.
- Can deal with bipartite graphs by looking at two step walks.
- Theorem: (Aleliunas, Karp, Lipton, Lovasz, Rackoff)

  The cover time $C(G) <= m* (2n-1)$

  Corollary: Undirected connectivity in $O(log\ n)$ space

  Corollary: Cover time for line graph is $O(n^2)$

Application to 2SAT

RWALK (randomized algorithm to test if 2CNF *F* is satisfiable)

Choose a random (or arbitrary) initial truth assignment tau

For *i = 1 .. c *n^2* (for *c* sufficiently large)

If *tau* satisfies *F*, report that a satisfying assignment has been found

Else find an unsatisfied clause and choose one of its literals *ell_i* at random. Change tau by flipping *ell_i*

End For

Claim: Let *tau** be a truth assignment satisfying *F*. Then we can view RWALK as a uniform random walk on a line graph (with nodes *0,1,…,n*) that is trying to reach node *n* where node *i* indicates that *tau* matches *tau** in *i* coordinates.

Better than 2^n for k-Sat

- Schoening utilizes this idea to show that for every k, there is a randomized algorithm with expected time O*(2(k-1)/k)^n for k-SAT.
- The idea is to start at a random tau and analyze the Prob[RWALK will reach tau*| conditioned on the initial tau being r from tau*]

3SAT analysis

- Schoening shows for every *k*, there is a randomized algorithm with expected time *O*(2(k-1)/k)^n* for *k*-SAT. For 3-Sat, start at a *random tau* and analyze the *Prob*[RWALK will reach *tau**| conditioned on the initial *tau* being *r* from tau*]. Consider walking *3r* steps. Prob of success is at least $\binom{r+2i}{i}\left(\frac{1}{3}\right)^{r+i}\left(\frac{2}{3}\right)^{i}$
- This is maximized at *i = r* so that the Prob of success is at least $\binom{3r}{r}\left(\frac{1}{3}\right)^{2r}\left(\frac{2}{3}\right)^{r}$
- To better understand this bound we need to estimate $\binom{3r}{r} = \frac{(3r)!}{r!(2r)!}$

Finishing the 3-SAT analysis

- Using Stirling's approximation for the factorial,
  $\binom{3r}{r}$ = $\Theta(\frac{(3^{3r}}{\sqrt{r}2^{2r}})$
  The probability then that we reach *tau** in *3r* steps is *Omega*(1/[sqrt{r} 2^r])* conditioned on the initial tau being r from tau*. The prob that the random tau will be distance r from tau* is $\sum_{r=0}^{n}\binom{n}{r}2^{-n}$.

The (unconditioned) probability will be at least

$$\Theta^*(\frac{1}{2^n}\sum_r \binom{n}{r}\frac{1}{2^r}) = \Theta^*(\frac{1}{2^n}(1+\frac{1}{2})^n) = \Theta^*(3/4)^n$$

- Using usual *(1-1/t)^t* bound with *t = O*[(4/3)^n]* shows that we can get constant probability within time *O*[(4/3)^n]*

Primality Testing

- I now want to turn attention to one of the most influential randomized algorithms, namely a poly time randomized algorithm for primality (or perhaps better called compositeness) testing.
- History of polynomial time algorithms:
  - 1-sided error with prob[ALG says *N* composite|*N* prime] = 0; prob[ALG say *N* prime|*N* composite] *<= delta* < *1*. Can then repeat.
  - Independently shown by Solovay and Strassen, and Rabin ~ 1974
  - The Rabin test is related to an algorithm by Miller ~1976 that gives a det poly time alg assuming (the unproven) ERH
  - 0-sided error alg (expected poly time) by Goldwasser and Kilian ~1986
  - deterministic poly time alg by Agarwal, Kayal and Saxena ~ 2002

- Even though there is a deterministic alg, it is not nearly as efficient as the 1-sided error algs which are used in practice and which also spurred the interest in this topic, had a major role in various cryptographic developments (which required random primes) and more generally became the impetus for the major interest in randomized algorithms.