Today

- Why PRGs?
- The use of PRGs in randomized algorithms
- Introduce expander graphs
- Random walks on expanders and Impagliazzo-Zuckerman PRG
Expander Graphs

- We will study expanders a lot the next few weeks.
- Constant degree (regular typically), constant conductance.
- By Cheeger we saw that we can characterize expanders through eigenvalues.
- A family of graphs is expanding if for $i > 1$:
  \[ |\lambda_i - d| \leq \epsilon d \text{ or } |\mu_i| \leq \epsilon d \]
Why Study PRGs?

- Pseudo-random number generators take a seed which is presumably random and generate a long string of random bits that are supposed to act random.
- Why would we want a PRG?
  - Random bits are scarce (e.g., low-order bits of temperature of the processor in a computer is random, but not too many such random bits). Randomized algorithms often need many random bits.
  - Re-run an algorithm for debugging, convenient to use same set of random bits. Can only do that by re-running the PRG with the same seed, but not with truly random bits.
Why Study PRGs?

- Standard PRGs are terrible (e.g. rand in C). Often produce bits that behave much differently than truly random bits.

- One can use cryptography to produce such bits, but much slower
Repeating an Experiment

• Consider wanting to run the same randomized algorithm many times.
• Let A be the algorithm, which returns “yes”/“no” and is correct 99% of the time (correctness function of the random bits)
• Boost accuracy by running A t times and taking majority vote
• Use truly random bits the first time we run A and then with the PRG we will see that every new time we only need 9 random bits.
• If we run t times, probability that majority answer is wrong is exponential in t.
The Random Walk Generator

- Let \( r \) be the number of bits out algorithm needs for each run: space of random bits is \( \{0,1\}^r \)
- Let \( X \subseteq \{0,1\}^r \) be the settings of random bits on which algorithm gives wrong answer for specific input.
- Let \( Y = \{0,1\}^r \setminus X \) be the settings on which algorithm gives the correct answer.
The Random Walk Generator: Expander Graphs

- Our PRG will use a random walk on a d-regular G with vertex set \{0,1\}^r, and degree d = constant.

- We want G to be an expander in the following sense: If \(A_G\) is G’s adjacency matrix and \(d = \alpha_1 > \alpha_2 \geq \cdots \geq \alpha_n\) its eigenvalues then we require that
  \[
  \frac{|\alpha_i|}{d} \leq \frac{1}{10}
  \]

Such graphs exist with d=400 (next weeks)
The Random Walk Generator

- For the first run of algorithm, we require $r$ truly random bits. Treat those bits as vertex of expander $G$.
- For each successive run, we choose a random neighbor of the present vertex and feed the corresponding bits to our algorithm.
- I.e, choose random $i$ between 1 and 400 and move to the $i$-th neighbor of present vertex. Need $\log(400) \sim 9$ random bits.
- Need concise description, don’t want to store the whole graph (e.g. see hypercube)
The Random Walk Generator

$G \quad \nu_0 \in \{0,1\}^r$

$t=0$
The Random Walk Generator

\[ G \]

\[ v_1 \in N(v_0) \quad t=1 \]
The Random Walk Generator

\[ G \]

\[ v_2 \in N(v_1) \]
The Random Walk Generator

\[ G \]

\[ v_3 \in N(v_2) \]

\[ t=3 \]
The Random Walk Generator
Assume we will run the algorithm \( t+1 \) times. Start with truly random vertex \( u \) and take \( t \) random walk steps.

Recall that \( X \) is the set of vertices on which the algorithm is not correct, we assume that \( |X| \leq \frac{2^r}{100} \) (algorithm correct 99% of time)

If at the end, we report the majority of the \( t+1 \) runs of algorithm, then we will return the correct answer as along as the random walk is inside \( X \) less than half the time.
We will show that
\[ \Pr[|S| > \frac{t}{2}] \leq \left(\frac{2}{\sqrt{5}}\right)^{t+1} \]

T={0,...,t} time steps
S={i: \nu_i \in X}
Formalizing the Problem

- Initial distribution is uniform (start with truly random string): $p_0 = 1/n$
- Let $\chi_X$ and $\chi_Y$ the characteristic vectors of $X$ and $Y$.
- Let $D_X = diag(X)$ and $D_Y = diag(Y)$
- Let $W = \frac{1}{d}A$ (not lazy) random walk matrix, with eigenvalues $\omega_1, \ldots, \omega_n$ such that $\omega_i \leq \frac{1}{10}$ by the expansion requirement.
- Want to show $\Pr[|S| > t/2] \leq \left(\frac{2}{\sqrt{5}}\right)^{t+1}$
The Probability to be in X

- Fix a set $R \subseteq \{0, \ldots, t\}$ of time steps.
- The probability that the walk is in $X$ exactly during the steps in $R$ is
  $$\Pr[W \text{ walk in } X \text{ exactly for } i \in R] = \langle 1, D_{Z_t}W \ldots WD_{Z_0}p_0 \rangle$$
- Where $Z_i = X$ if $i \in R$ and $Y$ otherwise
- Show that this probability is $\left(\frac{1}{5}\right)^{|R|}$.
- $\Pr[|S| > t/2] \leq \left(\frac{2}{\sqrt{5}}\right)^{t+1}$ follows.
The Proof

- **Claim.**
  \[
  \Pr[W \text{ walk in } X \text{ exactly for } i \in R] = \langle 1, D_{Z_t}W \ldots WD_{Z_0}p_0 \rangle = \left(\frac{1}{5}\right)^{|R|}
  \]

- **Lemma.**
  \[
  \|D_XW\| \leq 1/5.
  \]