2nd Moment Method

- Chebyshev's Inequality
  \[ \forall \lambda > 0 \quad \Pr \left( |X-\mu| > \lambda \sigma \right) \leq \frac{1}{\lambda^2} \]

- Another version
  \[ \Pr (X=0) \leq \frac{\text{Var}(X)}{E(X)^2} \]

- Pr(\text{no class Monday})
- Project proposal due Monday

Another 2nd moment inequality

\[ \Pr (X > 0) \geq \frac{(E(X))^2}{E(X^2)} \]

Follows from Cauchy-Schwartz

\[ \left[ E(XY) \right]^2 \leq E(X^2) E(Y^2) \]

Set \( Y = 1_{X>0} \)

\[ E(X) \leq E(X^2)^{\frac{1}{2}} E \left[ \left( 1_{X>0} \right)^2 \right] \]

Proof:

wlog \( E(X^2) > 0 \)

Let \( U = \frac{X}{\sqrt{E(X^2)}} \) \( V = \frac{Y}{\sqrt{E(Y^2)}} \)

\[ 2 |UV| \leq U^2 + V^2 \]

\[ \Rightarrow 2 |E(UV)| \leq E(UV) \leq E(U^2) + E(V^2) = 2 \]

\[ \Rightarrow \left| E(UV) \right|^2 \leq 1 \]

\[ \Rightarrow \left[ E(XY) \right]^2 \leq E(X^2) E(Y^2) \]
Lovász Local Lemma

Let $E_1, E_2, \ldots, E_n$ be a set of "bad" events.

Say we want to show $\Pr(\bigcap_{i=1}^n \overline{E_i}) > 0$ "positive probability that nothing bad happens."

2 cases where easy:
1. $E_i$ are mutually independent $(1-p)^n$
2. $\sum_{i=1}^n \Pr(E_i) < 1$ union bound suffice

LLL is clever comb

\[ \Pr(\bigcap_{i=1}^n \overline{E_i}) = \Pr(\overline{E}) \]

A dependency graph for $E_1, E_2, \ldots, E_n$ is $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ & $E_i$ is mutually indep. of $\{E_j \mid (i, j) \in E\}$

Lovász Local Lemma

Let $E_1, \ldots, E_n$ be a set of events s.t.
1. $\Pr(E_i) < p \quad \forall i$
2. The max degree in dependency graph is $d$
3. $4dp \leq 1$

Then $\Pr(\bigcap_{i=1}^n \overline{E_i}) > 0$

Several variants & generalizations (see notes)
Let $\varphi$ be a $k$-SAT formula with $n$ variables and $m$ clauses. Each clause has $k$ literals.

$$\Pr(\text{random assignment to variables satisfies a particular clause}) = 1 - \frac{1}{2^k}$$

$$\Pr(\exists \text{ unsatisfied clause}) \leq m \frac{1}{2^k}$$

If $m < 2^k \Rightarrow \exists$ satisfying assignment.

**Thm:** Let $\varphi$ be a $k$-SAT formula with $n$ variables and $m$ clauses. If no variable appears on $\geq \frac{T}{m} = \frac{2^k}{4k}$ clauses, then the formula has satisfying assignment.

**Pr:** Let $E_i$ be the event that clause $i$ is not satisfied.

$$p = \Pr(E_i) = 2^{-k}$$

$E_i$ is mutually independent of any clause it doesn't share variables with.

$$d \leq kT = k \cdot \frac{2^k}{4k} = \frac{2^k}{4}$$

$$4dp = 4 \cdot \frac{2^k}{4} \cdot 2^{-k} \leq 1 \Rightarrow \text{valid}.$$
Application 2: Packet Routing

A graph $G$ with $n$ packets, each packet has a source $s_i$ and destination $t_i$, and specific path $P_i: s_i \rightarrow t_i$.

Only one packet can traverse an edge per time unit.

Schedule specifies for each packet when to move, when to wait:

\[ d = \max\{P_i\} \quad \text{dilation} \]
\[ c = \max_{e} (\# \text{paths } P_i \text{ that use } e) \quad \text{congestion} \]

How long for each packet to reach its destination?

\[ N(c+d) \quad ??? \quad O(cd) \]

[Leighton, Rao, Maggs] $\exists$ schedule of length $O(c+d)$ always independent of $n$.

Can be proved using LLL

High level idea:

For each packet, assign random initial delay in $[1, \alpha(c+d)]$.

Guarantees limited dependency between congestion on different edges in different time periods.
Algorithmic version

2 clauses $C_i$ & $C_j$ are dependent if they share a van

$$D(C_i) = \{ C_j : C_i \& C_j \text{ dependent} \}$$

Let $d = \max |D(C_i)|$

**Theorem**

Let $\Phi$ be a $k$-SAT formula with $d \geq \frac{2^k}{3}$ (in clauses, not vars)

Then $\Phi$ is satisfiable & a satisfying assignment can be found in poly time.

---

**Algorithm**

Initialize $x^0 = (x_1, \ldots, x_n)$

where $x_i = \begin{cases} 1 & \text{w.p. } \frac{1}{k} \\ -1 & \text{w.p. } \frac{2}{k} \end{cases}$

While some clause $C$ that is not satisfied

Fix($C$)

Always process clauses in fixed order

---

**Fix($C$)**

Randomly reassign $k$ vars on $C$ to $+/-$ (indep w.p. $\frac{1}{2}$)

$\Rightarrow$ give updated $x$

While some clause $D \in \Phi$ that shares vars w/ $C$ is violated

Fix($D$)

**Observations**

1. #random bits used is $n + k \cdot \# \text{calls to Fix}$

2. If Fix($C$) terminates, then it terminates with assignment in which all clauses in $D(C)$ are satisfied.

3. $S = \{ S_i \text{ satisfied by previous level call to Fix} \}$

   - $S_i$ satisfied by previous level call to Fix,$(i)$

   If Fix($C_i$) terminates, then all clauses in $S$ are still satisfied.

$\Rightarrow$ make progress when all other calls finish.
Then \( V \text{ kSAT formula } u \vdash d \leq 2^k \) alg terminates in polytime w.h.p.

**Proof:** \( f : A \rightarrow B \) injective \( |B| \geq |A| \)

Suppose abort the computation after \( T \) calls to \( \text{Fix} \)

ALG uses up to \( n + kT \) bits.

Let \( A = \{0, 1\}^{n+kt} \)

\(|A| = 2^{n+kt} \)

Write down transcript of computation for fixed \( x_0, y_0 \)

\( x_0, y_0, z \xrightarrow{\text{Fix}(c_i)} x_1, y_1, z_1, \quad x_1, y_1, z_2 \xrightarrow{\text{Fix}(c_i)} x_2, y_2, z_2 \rightarrow \ldots \rightarrow x_T, e, z_T \)

\( \uparrow \)

- \( z' \) is obtained from \( z \) by appending
  - 1 binary rep of \( C_i \) if outer call to \( C_i \): \( \left\lceil \log_2(n) \right\rceil + 1 \)
  - 1 "binary rep of \( C_i \)" if inner call to \( C_i \): \( \left\lceil \log_2(n) \right\rceil + 1 \) bits.
Claim: transcript is reversible

1. Construct

\[ f(x_0, y_0, z) \rightarrow (x_1, y_1, z_1) \]

\[ n + kT \text{ bits} \rightarrow n + \frac{\text{bits for input cells}}{m(\log_2(n)+2)} + \frac{\text{bits for inner cells}}{T(\log_2(d)+2)} \leq k-3 \]

\[ n + kT \rightarrow \frac{\text{bits in final transcript}}{2} \leq n + m(\log_2(n)+2) + T(k-1) \]

All G doesn't terminate

\[ #\text{inputs} = #\text{outputs} \]

\[ n + kT \leq n + m(\log_2(n)+2) + T(k-1) \]

\[ \Rightarrow T \leq m(\log_2(n)+2) \]
\[ \text{If } T > S \implies \exists \text{ input on which ALG terminates} \]

Suppose that on fraction \( \geq 2^{-c} \) of inputs ALG doesn't terminate.

\[
2^{n+k-c} \leq \frac{\# \text{ inputs on which it doesn't terminate}}{\text{ALG terminates}} \leq 2^{n+m(S \log_2(n)+2) + T(k-1)}
\]

\[
\implies T \leq \frac{m(S \log_2(n)+2)}{S} + 5
\]

If \( T > S \), then alg doesn't terminate on \( < 2^{-c} \) fraction of inputs.

Succeed w.p. \( \geq 1 - 2^{-c} \).