CS787: Advanced Algorithms	**Scribe:** Group 4
Title: hw2	

0.1 Max Cut

(A) $|F|/2$. For each edge in $|F \cap (V_1 \times V_2)|$, it's two vertices need to be in different partition. Because every vertex is independent, so the probability of that is 1/2*1/2*2 = 1/2. Thus expected number of edges cut is $|F|/2$.

(B) As mentioned in (A), every edge has 1/2 probability to be cut, and has 1/2 probability not to be cut. It's just like the "flipping n coins" problem.
Let the cut edges set be $|C|$.
Chebyshev's Inequality provides an upper bound:
$Pr(|C| < |F|/4) \leq \frac{\sigma^2}{t^2} = \frac{(|F|/4)}{(|F|/4)^2} = \frac{4}{|F|}$

Chernoff-Hoeffding Bounds, here $u = |F|/2$ and $\delta = 1/2$:
$Pr(|C| < |F|/4) \leq e^{-|F|/16}$

If $e^{-\delta^2 \frac{|F|}{4}} = \frac{4}{|F|}$, $\delta = \sqrt{\frac{4(ln|F|-ln4)}{|F|}}$, the probability of $|F|$ be fewer than $\frac{|F|}{2} - \frac{|F|}{2}\sqrt{\frac{4(ln|F|-ln4)}{|F|}}$ will be very small.

0.2 Chernoff Bound

(A) We apply log operation since log do not change the comparison result of the left hand side to the right hand side. Then, applying Arithmetic-Geometric Mean Inequality should complete the proof.

$$\ln \frac{f(a)+f(b)}{2} = \ln \frac{e^{ta}+e^{tb}}{2} \geq \ln(e^{ta}e^{tb})^{0.5} = \frac{\ln e^{ta} + \ln e^{tb}}{2} = \frac{ta+tb}{2} = \ln e^{t(\frac{a+b}{2})}$$
$$= \ln f(\frac{a+b}{2})$$

Thus, $\frac{f(a)+f(b)}{2} \geq$ f$(\frac{a+b}{2})$

(B) Let us denote $f_C(x)$ be the probability distribution function of C, and E[C] = E[B] = μ, we have:

$$
\begin{aligned}
E[f(C)] &= \int_0^1 c * f_C(c)dc \\
&\leq \int_0^1 f_C(c)[(1-c)*f(0) + c*f(1)]dc \quad (Since\ f(x)\ is\ convex\ in\ [0,1]) \\
&= f(0)\int_0^1 f_C(c)dc + (f(1)-f(0))\int_0^1 c*f_C(c)dc \\
&= f(0) + \mu(f(1)-f(0)) \\
&= (1-\mu)f(0) + \mu f(1) \\
&= E[f(B)]
\end{aligned}
$$

(C) Let us denote:
x_i be a continuous random variable in [0,1], and X = $\sum_i x_i$
y_i be a discrete random variable in $\{0,1\}$, and Y = $\sum_i y_i$
P_i = Pr(y_i=1), and μ = E[Y] = E[$\sum_i y_i$] = $\sum_i P_i$

We need to prove for any $\delta > 0$, we have
upper tail: $Pr[X \geq (1+\delta)\mu] \leq (\frac{e^\delta}{(1+\delta)^{(1+\delta)}})^\mu$
lower tail: $Pr[X \leq (1-\delta)\mu] \leq (\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}})^\mu$

Given a increasing f(x), the Markov's Inequality gives us

$$Pr[f(x) \geq f(\lambda)] \leq \frac{E[f(x)]}{f(\lambda)}$$

To prove the upper tail, we choose f(x) = e^{xt} for some t > 0,

$$
\begin{aligned}
Pr[X \geq (1+\delta)\mu] = Pr[f(x) \geq f(\lambda)] &\leq \frac{E[e^{Xt}]}{e^{(1+\delta)\mu t}} = \frac{E[e^{\sum_i x_i t}]}{e^{(1+\delta)\mu t}} = \frac{\prod_i E[e^{x_i t}]}{e^{(1+\delta)\mu t}} \leq \frac{\prod_i E[e^{y_i t}]}{e^{(1+\delta)\mu t}} \\
&= \frac{\prod_i (1+P_i(e^t-1))}{e^{(1+\delta)\mu t}} \leq \frac{\prod_i e^{P_i(e^t-1)}}{e^{(1+\delta)\mu t}} = \frac{e^{\mu(e^t-1)}}{e^{\mu(1+\delta)t}} = (\frac{e^{e^t-1}}{e^{(1+\delta)t}})^\mu
\end{aligned}
$$

$(\frac{e^{e^t-1}}{e^{(1+\delta)t}})^\mu$ gets minimum value $(\frac{e^\delta}{(1+\delta)^{(1+\delta)}})^\mu$ when t = $\ln(1+\delta)$, thus the upper tail is proved.

To prove the lower tail, we choose a decreasing function $f(x) = e^{-xt}$ for some t > 0.

$$
\begin{aligned}
Pr[X \leq (1-\delta)\mu] = Pr[f(x) \geq f(\lambda)] &\leq \frac{E[e^{-Xt}]}{e^{(1-\delta)\mu t}} = \frac{E[e^{\sum_i -x_i t}]}{e^{(1-\delta)\mu t}} = \frac{\prod_i E[e^{-x_i t}]}{e^{(1-\delta)\mu t}} \leq \frac{\prod_i E[e^{-y_i t}]}{e^{(1-\delta)\mu t}} \\
&= \frac{\prod_i (1+P_i(e^{-t}-1))}{e^{(1-\delta)\mu t}} \leq \frac{\prod_i e^{P_i(e^{-t}-1)}}{e^{(1-\delta)\mu t}} = \frac{e^{\mu(e^{-t}-1)}}{e^{\mu(1-\delta)t}} = (\frac{e^{e^{-t}-1}}{e^{-t(1-\delta)}})^\mu
\end{aligned}
$$

$(\frac{e^{e^{-t}-1}}{e^{-t(1-\delta)}})^\mu$ gets minimum value $(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}})^\mu$ when t = -$\ln(1-\delta)$, thus the lower tail is proved.

0.3 Longest Path

(A) Color(v) is the color of the node v, Sets(v, d) is the sets of colors of different paths which end with node v and has length d.
Our algorithm:
1. initialize $Sets(s, d) = \{Color(s)\}$, $Sets(v, 1) = \emptyset$, $v \neq s$
2. For i from 1 to k-1
3. For each edge (u, v)
4. For each SET in Sets(u, i)
5. If Color(v) is not in SET
6. $Sets(v, i + 1) = Sets(v, i + 1) \cup (SET \cup \{Color(v)\})$
7. If $Sets(t, k) = \emptyset$ return false, else return true.

Prove:
Firstly, we loop i from 1 to k-1. Secondly, we traversed all the edges $|E|$. Thirdly, we traverse all Sets of Colors for the node u, and the MAX number of sets of colors of length i is $\binom{k}{i}$. And we assume that SET is a hash set, then time complexity of accessing it will be $O(1)$. Thus the running time will be $O(\sum_{i=1}^{k} |E|\binom{k}{i}) = O(2^k|E|)$, so this problem is FPT with parameter k.

(B) If we randomly color nodes using our algorithm in part (A), we will have k^k ways to color k nodes, but k! ways to color k nodes with k different colors, so the probability for the optimal path is colorful will be:
Pr(optimal is colorful) $= \frac{k!}{k^k} >= e^{-k}$
Then Pr(optimal is not colorful) $<= 1 - e^{-k}$
If we run T times, because we want to make sure that Pr(optimal is colorful after run n times) $>= 1 - \frac{1}{poly(n)}$.
Which means Pr(optimal is not colorful after run T times) = Pr(optimal is not colorful)^T $<= (1 - e^{-k})^T <= \frac{1}{poly(n)}$
So $T = log_{\frac{e^k}{e^k - 1}} poly(n) = O(poly(n))$. $(k >= 1)$
Because the time complexity of Our algorithm in part (A) is $O(2^k|E|)$, thus if run T times, time complexity will be: $O(T * 2^k|E|) = O(2^{O(k)}poly(n))$.

0.4 Dual Fitting for Set Cover

(A) n is the number of total elements.

$$max \sum_{i=1}^{n} y_i \quad s.t.$$
$$\sum_{i: e_i \in S_j} y_i \leq c_j, \forall j$$
$$y_i \geq 0, \forall i$$

(B) We directly use dual fitting to prove the weighted version in (C). Thus, the proof in (C) should be also applicable here since the unweighted version is just a special case of the weighted version when all c_j are equal to 1.

(C) Let us define some notations:

ALG: the proposed greedy algorithm
l: the number of total iterations that ARG takes
$\hat{S}_j$: the set of uncovered elements in S_j at the start of a particular interation
g: the maximum size of all subsets, that is g = $max|S_j|$
I: the indices of the subsets in final ALG solution
OPT: the optimal integral solution
P: the primal linear programming solution
D: the dual linear programming solution

First, we construct an infeasible dual solution such that $\sum_{i=1}^{n} y_i = \sum_{j \in I} c_j$ = cost(P). (We will show why it is infeasible after proving the second)
Second, we will prove that y' = $\frac{1}{H_g} y$ is a feasible dual solution.
By the weakly duality theorem, we get:

$$\begin{aligned} cost(ALG) &= \sum_{j \in I} c_j = \sum_{i=1}^{n} y_i = H_g \sum_{i=1}^{n} y'_i \leq H_g cost(P) \leq H_g cost(OPT) \leq H_n cost(OPT) \\ &= O(ln(n)) cost(OPT) \end{aligned}$$

Below inequality shows H_n is θ(ln(n)) (Hence also O(ln(n))):

$$\sum_{x=1}^{n} \frac{1}{x+1} \leq \int_{1}^{n} \frac{1}{x} dx = ln(n) \leq \sum_{x=1}^{n} \frac{1}{x}$$

Now we prove y'=$\frac{1}{H_g} y$ is feasible:
We must show that for each subset S_j, $\sum_{i: e_i \in S_j} y'_i \leq c_j$. Pick an arbitrary subset S_j. Let a_k be the number of elements in this subset that are still uncovered at the beginning of kth iteration, so that $a_1 = |S_j|$ and $a_{l+1} = 0$. Let A_k be the uncovered elements of S_j covered in the kth iteration, so that $|A_k| = a_k - a_{k+1}$. If subset S_p is chosen in the kth iteration, then for each element $e_i \in A_k$ covered in the kth iteration, we have:

$$y'_i = \frac{c_p}{H_g |\hat{S}_p|} \leq \frac{c_j}{H_g a_k}$$

The inequality follows because if S_p is chosen in the kth iteration, it must minimize the ratio of its weight to the number of uncovered elements it contains. Thus,

$$
\begin{aligned}
\sum_{i:e_i \in S_j} y'_i &= \sum_{k=1}^{l} \sum_{i:e_i \in A_k} y'_i \\
&\leq \sum_{k=1}^{l} (a_k - a_{k+1} \frac{c_j}{H_g a_k}) \\
&\leq \frac{c_j}{H_g} \sum_{k=1}^{l} (\frac{1}{a_k} + \frac{1}{a_k - 1} + ... + \frac{1}{a_{k+1} + 1}) \\
&\leq \frac{c_j}{H_g} \sum_{i=1}^{|S_j|} \frac{1}{i} \\
&= \frac{c_j}{H_g} H_{|S_j|} \\
&\leq c_j
\end{aligned}
$$

Hence, we complete the proof.
Now given the result of the proof, we can derive:

$$
\sum_{i:e_i \in S_j} y_i = H_g \sum_{i:e_i \in S_j} y'_i \leq H_g c_j
$$

meaning a certain dual constrain can be violated, implying y might be infeasible.