Q1

(a) For each edge e, let X_e be the indicator random variable for e such that $X_e = 1$ if it is cut and $X_e = 0$ otherwise. Then for each edge e, $\mathbf{E}[X_e] = \mathbf{Pr}[X_e = 1] = \frac{1}{2}$. Denote the number of edges cut by X. Then $X = \sum_{e \in F} X_e$. By the linearity of expectation, $\mathbf{E}[X] = \sum_{e \in F} \mathbf{E}[X_e] = \frac{|F|}{2}$.

(b) For each e, the variance is $\mathbf{Var}[X_e] = \frac{1}{4}$. Since the set of indicator variables $\{X_e\}_{e \in F}$ are *pair-wise independent*, we have $\mathbf{Var}[X] = \sum_{e \in F} \mathbf{Var}[X_e] = \frac{|F|}{4}$. Applying Chebyshev's inequality, $\mathbf{Pr}\left[X \le \frac{|F|}{4}\right] \le \mathbf{Pr}\left[|X - \frac{|F|}{2}| \ge \frac{|F|}{4}\right] \le \frac{\mathbf{Var}[X]}{(|F|/4)^2} = \frac{4}{|F|}$.

Note that the set of $\{X_e\}_{e \in F}$ are **not** *mutually independent* (imagine a cycle of three edges), so one cannot apply Chernoff-Hoeffding Inequality.

Q2

(a) For every $a, b \in \mathbb{R}$, $\frac{f(a)+f(b)}{2} - f(\frac{a+b}{2}) = \frac{e^{ta}+e^{tb}}{2} - e^{t(a+b)/2} = \left(\frac{e^{ta/2}}{\sqrt{2}} - \frac{e^{tb/2}}{\sqrt{2}}\right)^2 \ge 0$.

(b) Let X be a random variable such that $X = a$ with probability $1-\lambda$ and $X = b$ with probability λ. According to Jensen's inequality, for any $\lambda \in [0, 1]$ we have

$$f(\mathbf{E}[X]) = f((1-\lambda)a + \lambda b) \le (1-\lambda)f(a) + \lambda f(b) = \mathbf{E}[f(X)].$$

Write $C = (1 - C) \cdot 0 + C \cdot 1$, then

$$\begin{aligned}
\mathbf{E}[f(C)] &= \mathbf{E}[f((1-C) \cdot 0 + C \cdot 1)] \\
&\le \mathbf{E}[(1-C) \cdot f(0) + C \cdot f(1)] && \text{(Jensen's inequality)} \\
&= (1 - \mathbf{E}[C]) \cdot f(0) + \mathbf{E}[C] \cdot f(1) && \text{(linearity of expectation)} \\
&= (1 - \mathbf{E}[B]) \cdot f(0) + \mathbf{E}[B] \cdot f(1) && (\mathbf{E}[B] = \mathbf{E}[C]) \\
&= \mathbf{Pr}[B = 0] \cdot f(0) + \mathbf{Pr}[B = 1] \cdot f(1) && (\mathbf{E}[B] = \mathbf{Pr}[B = 1]) \\
&= \mathbf{E}[f(B)].
\end{aligned}$$

(c) For each random variable $X_i \in [0, 1]$, let $Y_i \in \{0, 1\}$ be a random variable such that $\mathbf{Pr}[Y_i = 1] = \mathbf{E}[X_i]$. Thus $\mathbf{E}[Y_i] = \mathbf{E}[X_i]$ and if we let $Y = \sum_{i=1}^{n} Y_i$ then $\mathbf{E}[Y] = \mathbf{E}[X]$. Write $\mathbf{Pr}[Y_i = 1] = P_i$ for short. Let $f(x) = e^{tx}$ for some $t > 0$ to be decided later. Then by part (b) we know $\mathbf{E}[f(X_i)] \le \mathbf{E}[f(Y_i)]$.

Then the following is essentially a repetition of what we learn in class. We only show one direction here.

$$
\begin{aligned}
\mathbf{Pr}[X \geq (1+\delta)\mu] &= \mathbf{Pr}[f(X) \geq f((1+\delta)\mu)] \\
&\leq \frac{\mathbf{E}[f(X)]}{e^{(1+\delta)\mu t}} && \text{(Markov's inequality)} \\
&= \frac{\mathbf{E}\left[f(\sum_{i=1}^{n} X_i)\right]}{e^{(1+\delta)\mu t}} \\
&= \frac{\prod_{i=1}^{n} \mathbf{E}[f(X_i)]}{e^{(1+\delta)\mu t}} && \text{(mutual independence)} \\
&\leq \frac{\prod_{i=1}^{n} \mathbf{E}[f(Y_i)]}{e^{(1+\delta)\mu t}} \\
&= \frac{\prod_{i=1}^{n} (1 + P_i(e^t - 1))}{e^{(1+\delta)\mu t}} \\
&\leq \frac{\prod_{i=1}^{n} e^{P_i(e^t-1)}}{e^{(1+\delta)\mu t}} && (1 + x \leq e^x) \\
&= \frac{e^{(e^t-1)\mu}}{e^{(1+\delta)\mu t}} \\
&= \left(\frac{e^{(e^t-1)}}{e^{(1+\delta)t}}\right)^{\mu}.
\end{aligned}
$$

The analysis for optimizing the bound by choosing appropriate t would be similar to the $\{0,1\}$ variable case.

Q3

(a) Because we are asked to find if there exists a colorful path with at least k intermediate nodes and there are k colors in total, the target paths must have exactly k intermediate nodes. We use dynamic programming to give the FPT algorithm.

Let $\text{HasPath}(C, v)$ denote if there is a colorful path from s to v using colors in the set C. In other words, $\text{HasPath}(C, v)$ is true if there is one and is false if there is none. Then we can write a recurrence relation as follows:

$$
\text{HasPath}(C, v) = \bigvee_{u \in N(v)} \text{HasPath}(C \setminus \{c_u\}, u)
$$

where $N(v)$ denotes the neighbors of vertex v and c_u is the color of vertex u. The base cases are $\text{HasPath}(\emptyset, v)$ being true and $\text{HasPath}(\emptyset, w)$ being false for every other vertex $w \neq v$.

This dynamic programming problem HasPath has $2^k \cdot n$ entries and filling each entry takes $O(n)$ time. Therefore, in $O(2^k n^2)$time we can compute the target value HasPath$([k], t)$. Hence the problem is FPT.

(b) Our algorithm runs t rounds where t is to be determined later:

(I) Initialize $i = 0$

(II) Uniformly randomly color each vertex

(III) Run the FPT algorithm in part (a) and return the k-path if found

(IV) Let $i = i + 1$ and repeat (I) until $i = t$

(V) Report failure

In a single round, we give each vertex a uniformly random color (each with probability $\frac{1}{k}$). If there exists an k-path P between s and t, then we have $\mathbf{Pr}[P \text{ is a colorful } k\text{-path}] = \frac{k!}{k^k} \approx \frac{1}{e^k}$ (by Stirling's approximation).

If there exists a k-path in the graph, the probability that the FPT algorithm in part (a) cannot find a colorful k-path at this round is at most $1 - \frac{1}{e^k}$. If we run the above algorithm for t rounds, then the probability that all of the t rounds fail is at most $\left(1 - \frac{1}{e^k}\right)^t$.

If we set $t = e^k \log n$, then we have

$$\left(1 - \frac{1}{e^k}\right)^t = \left(\left(1 - \frac{1}{e^k}\right)^{e^k}\right)^{\log n} < \frac{1}{e^{\log n}} = \frac{1}{n}.$$

Therefore, the probability that there exists at least one k-path but our algorithm fails to find one is at most $\frac{1}{n}$. Combining the time bound in part (a), the algorithm runs in $O(2^{O(k)} n^2 \log n)$.

Q4

(a) Recall the program Primal:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} c_i x_i \\ \text{subject to} \quad & \sum_{i: e_j \in S_i} x_i \geq 1, \ \forall j \in [n] \\ & x_i \geq 0, \ \forall i \in [m] \end{aligned}$$

In the program DUAL a variable y_j is introduced for each edge e_j:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} y_j \\ \text{subject to} \quad & \sum_{j:e_j \in S_i} y_j \le c_i, \ \forall i \in [m] \\ & y_j \ge 0, \ \forall j \in [n] \end{aligned}$$

(b) Denote the number of subsets returned by the greedy algorithm by q and denote the subsets returned by $S_{\alpha_1}, \cdots, S_{\alpha_q}$ (with the order preserved). Let $F_0 = F$ and $F_p = F_{p-1} \setminus S_{\alpha_p}$ for each $1 \le p \le q$. Note that $F_q = \emptyset$. If an element e_j is first covered by the subset S_{α_p}, let $z_j = \frac{1}{|S_{\alpha_p} \cap F_{p-1}|}$. Consider $(z_1, \ldots, z_q)$ as an input to the program DUAL. The target value is

$$\begin{aligned} \sum_{j=1}^{m} z_j &= \sum_{p=1}^{q} \sum_{e_j \in S_{\alpha_p} \cap F_{p-1}} z_j \\ &= \sum_{p=1}^{q} \sum_{e_j \in S_{\alpha_p} \cap F_{p-1}} \frac{1}{|S_{\alpha_p} \cap F_{p-1}|} \\ &= \sum_{p=1}^{q} 1 \\ &= q \end{aligned}$$

Thus the dual solution $(z_1, \ldots, z_q)$ has exactly the same cost as the primal solution, but it might not satisfy the constraints for DUAL. Next we show it violates the dual constraints by a factor of at most H_n, i.e. $\sum_{j:e_j \in S_i} y_j \le 1, \ \forall i \in [m]$, and hence we can rescale this dual solution so that it satisfies the dual constraints exactly.

Consider the subset S_i. Denote the size of S_i by t and its elements by $e_{\beta_1}, \ldots, e_{\beta_t}$, in the order of being added into the set cover. Since each element in S_i is covered by the greedy algorithm, we know that before each e_{β_s} $(1 \le s \le t)$ is covered, the set S_i has at least $t - s + 1$ elements uncovered. Let us assume e_{β_s} is first covered in some subset S_{α_p}, so $z_{\beta_s} = \frac{1}{|S_{\alpha_p} \cap F_{p-1}|}$. We can infer that $|S_{\alpha_p} \cap F_{p-1}| \ge |S_i \cap F_{p-1}| = t - s + 1$ because otherwise we would have chosen S_i rather than S_{α_p} in this round. That is to say, $|S_i \cap F_{i-1}| \ge t - s + 1$ and consequently

$$z_{\beta_s} \le \frac{1}{t - s + 1}.$$

Therefore, we have

$$\sum_{j:e_j \in S_i} z_j = \sum_{s=1}^{t} z_{\beta_s} \le \sum_{s=1}^{t} \frac{1}{t - s + 1} = H_t \le H_n \approx \log n.$$

This means that $(z_1, \ldots, z_q)$ only violates the dual constraints by a factor of at most H_n and hence $(z_1/H_n, \ldots, z_q/H_n)$ is a feasible solution for the program DUAL and proves that the greedy algorithm gives an $O(\log n)$s approximation for the unweighted set cover problem.

(c) Proof for the weighted case is essentially the same as in part (b) except that we need to set $z_j = \frac{c_{\alpha_p}}{|S_{\alpha_p} \cap F_{p-1}|}$ rather than $\frac{1}{|S_{\alpha_p} \cap F_{p-1}|}$.