1. In class we presented an efficient randomized algorithm for bipartite matching on $d$-regular graphs.

   (a) What goes wrong if the graph is not $d$-regular?
   (b) Additionally, we showed that the algorithm achieves $O(n \log n)$ time in expectation. Show an algorithm that achieves $O(n \log n)$ time with high probability. **Hint:** it may help to recall how we showed that the coupon collector takes $O(n \log n)$ samples with high probability.

2. Consider the example given in class for how online bipartite matching using random edges achieves a competitive ratio of $R = 1/2$: each arriving vertex $x_i$ has an edge to $y_i$ as well as all of $y_{n/2}, \ldots, y_n$. Show that the algorithm that the algorithm given in class, which randomly ranks the right vertices $y_i$, has $R \leq 3/4$ on this example.

3. Suppose you can sample elements $x_1, \ldots, x_m$ from a mean zero subgaussian variable $X$, and would like to compute its kurtosis $\kappa = \frac{\mathbb{E}[X^4]}{\mathbb{E}[X^2]^2}$.

   (a) Consider the algorithm that computes the empirical fourth and second moments $\frac{\sum x_i^4}{m}$ and $\frac{\sum x_i^2}{m}$, and outputs the former divided by the latter squared. How large must $m$ be for this to be an $(\epsilon, \delta)$ approximation to $\kappa$?
   (b) Construct a different algorithm that achieves an $(\epsilon, \delta)$ approximation to $\kappa$ with fewer measurements.
4. For a vector $x \in \mathbb{R}^n$, let $H_{k,p}$ denote the $\ell_p$ heavy hitters:

$$H_{k,p} := \{ i \in [n] \mid x_i^p \geq \|x\|_p^p / k \}.$$ 

We say that an algorithm solves the $\ell_p$ heavy hitters problem if it returns a set $\tilde{H}$ for which

$$H_{k,p} \supseteq \tilde{H} \supseteq H_{2k,p}.$$ 

with probability $1 - \delta$. In class, we discussed the following algorithms:

- The Count-Min Sketch, which solves $\ell_1$ heavy hitters in $O(k \log(n/\delta))$ space and $O(n \log(n/\delta))$ time.
- The Count-Min Sketch Tree, which solves $\ell_1$ heavy hitters in $O(k \log^2(n/\delta))$ space and $O(k \log^2(n/\delta))$ recovery time.
- The Count-Sketch, which solves $\ell_2$ heavy hitters in $O(k \log(n/\delta))$ space and $O(n \log(n/\delta))$ recovery time.

In this problem we will construct a sublinear time algorithm for $\ell_2$ heavy hitters.

(a) Consider a partition $S_1, \ldots, S_m$ of $[n]$, and suppose that $u \in S_i$ is an $\ell_2$ heavy hitter, i.e. $x_u^2 \geq \|x\|_2^2 / k$. In contrast to the $\ell_1$ case, show that $i$ is not necessarily an $\ell_2$ heavy hitter for the vector $y \in \mathbb{R}^m$ given by

$$y_j := \sum_{v \in S_j} x_v.$$ 

(b) Show that $i$ has a constant chance of being an $\ell_2$ heavy hitter for the vector $y \in \mathbb{R}^m$ given by

$$y_j := \sum_{v \in S_j} \sigma(v) x_v.$$ 

where $\sigma : [n] \rightarrow \{\pm 1\}$ is a fully independent hash function. (If you would like, also prove this when $\sigma$ is $O(1)$-wise independent.)

(c) Adapt the count-min sketch tree construction into a sublinear time algorithm for $\ell_2$ heavy hitters, i.e. one that runs in $O(k \log^c(n/\delta))$ time and space for some constant $c$. 

2