Exercise 13

Random Edge Identifiers

(1a) Let $e, e' \in E$ and let $\mathcal{E}_{e,e'}$ be the (bad) event where $e$ and $e'$ are assigned the same identifier. By taking the union bound over all $\mathcal{E}_{e,e'}$, we get that
\[
\bigcup_{e,e' \in E} \mathcal{E}_{e,e'} \leq n^2 \cdot 2^{-10 \log n} \leq n^{-8}.
\]

(1b) In this exercise, a little care has to be taken, since the probability of $\oplus_{e' \in E'} I_{e'} = I_e$ is not (necessarily) independent of $I_e$ in case $e \in E'$. However, it is the case $\oplus_{e' \in E'} I_{e'} = I_e$ if $\oplus_{e' \in E'} I_{e'} = I_{e^{-1}}$, where $I_{e^{-1}}$ stands for a bit string where all bits of the identifier of $e$ are flipped. Now, let $\mathcal{E}_{e,E'}$ be the (bad) event that $\oplus_{e' \in E'} I_{e'} = I_e$ if $e \notin E'$ and $\mathcal{E}_{e,E'}$ analogously for the case of $e \in E'$. Similarly to (1a), we can use union bound to get
\[
\bigcup_{e,e' \in E} \{\mathcal{E}_{e,E'} \cup \mathcal{E}_{e,E'}^c\} \leq 2n^2 \cdot 2^{-10 \log n} \leq n^{-7}.
\]

Graph Sketching for Connectivity

(2a) The proof follows closely the steps in Lemma 1 in the lecture notes. Consider some connected component $A$, let $B = V \setminus A$ and let $k$ be the number of edges between $A$ and $B$. For some estimate $\hat{k}$ it holds that $\hat{k}/2 \leq k \leq 2\hat{k}$. It is known that $1 - x \geq 4^{-x}$, when $0 \leq x \leq 1/2$. In the phase of Boruvka’s algorithm, where estimate $\hat{k}$ is made, the probability of choosing exactly one edge between $A$ and $B$ is at least
\[
\frac{\hat{k}}{2} \cdot \frac{1}{\hat{k}} \left(1 - \frac{1}{\hat{k}}\right)^{2\hat{k}} \geq \frac{\hat{k}}{2} \cdot \frac{1}{\hat{k}} \left(4^{\frac{1}{2}}\right)^{2\hat{k}} \geq \frac{1}{40}.
\]

(2b) In every phase of the algorithm, we remove at least $1/80$ components in expectation. Setting the constant in the $O$ notation large enough, we can use the calculations from the previous exercises (Exercise 12) to obtain the result.

Graph Sketching for Testing Bipartiteness

Consider the following graph construction. We replace every node $v \in V$ with two nodes, $v_{in}$ and $v_{out}$ and connect $v_{in}$ to $v_{out}$ for every $\{v,u\} \in E$. Recall now, that a bipartite graph has no odd cycles. Furthermore, if there are only even cycles in the graph, any path from node $v_{in}$ leads back to $v_{in}$. This follows from the observation that any path has an even amount of steps and every second node on the path is going to be an “in” node. Conversely, a path from $v_{in}$ to $v_{out}$ can be found by following a path starting from $v_{in}$ going around an odd cycle back to $v_{out}$. Due to the odd amount of steps in this path, the end must be an “out” node. Therefore, we can test bipartiteness by validating that for all $v \in V$, $v_{in}$ is not connected to $v_{out}$.