Randomness and Computation 2018/19
Week 8 tutorial sheet (12-1pm, Tues 5th, Wed 6th March)

1. Given a graph $G = (V, E)$, a *vertex cover* of G is a set of vertices $C \subseteq V$ such that each edge has at least one endpoint in C. Finding the vertex cover of the smallest cardinality is NP-complete (we do not expect to find any polynomial-time algorithm to compute a minimum cover for an arbitrary graph G) .

   Consider the following algorithm for Vertex Cover:

   i. Start with $C \leftarrow \emptyset$.
   ii. Pick an edge (u, v) such that $\{u, v\} \cap C = \emptyset$. Add an arbitrary endpoint (u or v) to C.
   iii. If C is a vertex cover, halt and return C, else continue at Step ii.

   We assume the graph has no isolated vertices (all vertices have some adjacent edge(s)).

   (a) Give an instance on which this algorithm may return a set which is $\Omega(n)$ times worse than the smallest vertex cover.

   (b) Now suppose we randomize the algorithm: when we choose an edge (u, v), we flip an unbiased coin to decide which endpoint to add to C. If k is the size of a smallest vertex cover, show that $\mathrm{E}[|C|] \leq 2k$.

   Note: The choice of edge (u, v) itself is not assumed to be randomized, just the choice of the endpoint.

   (c) The result of part (b) only guarantees the expectation satisfies the claimed bound. However we have the option of re-running the algorithm to get alternative vertex covers. Show how we can set a value ℓ for "number of re-runs" of the algorithm, which is large enough to ensure that with probability at least $(1 - \delta)$ one of the results is a cover of size $\leq 2k$ (for an arbitrary given $\delta > 0$).

   (d) Suppose each vertex v has an associated positive weight $w(v)$, and the objective is to pick a vertex cover of smallest possible weight (so $\sum_{v \in C} w(v)$ is the minimum possible subject to covering the graph). Give an example to show that the above algorithms do not work for this problem.

   (e) Now alter the modified algorithm of (b) to deal with the "weights" problem: after choosing an edge (u, v), add u to the cover with probability $\frac{w(v)}{w(u)+w(v)}$. If W is the weight of a least-weight vertex cover for G, w, show that this new algorithm returns a cover with *expected* total weight $\leq 2W$.

2. Exercise 5.16 on "cliques, $K_{3,3}$ and Hamilton cycles in $G_{n,p}$" from [MU].

   You are asked to consider these three structures (one by one) in the $G_{n,p}$ model and show how $p = p(n)$ should be set if we want to ensure that the *expected number* of the particular structure will be exactly 1 (note that to analyse this case is unusual; we are usually focusing on the case when expectation is > 0, maybe the case of high expected value).

   Note that all solutions will make use of the *linearity of expectation*

3. Exercise 6.9 on "tournaments" from [MU].

   We first give the definition of a *tournament* graph, a graph on n vertices which has exactly one directed edge between each pair of vertices. We also define the concept of a *ranking*, this being an ordering/permutation of the n-items from best to worst (no ties allowed). We say that a ranking *disagrees* with a directed edge $y \to x$ if y is ahead (better than) of x in the ranking.

   We are interested in the question, for a given tournament graph, or whether there exist rankings which are close (in both directions) to matching half of the tournament.

   (a) Prove that, for every tournament, there exists a ranking that disagrees with at most 50% of the edges.

   (b) Prove that, for sufficiently large n, there exists a tournament such that *every* ranking disagrees with at least 49% of the edges in the tournament.

   The second part of this question is difficult (requiring a complicated/creative use of Chernoff bounds among other things); don't be worried if you don't solve it alone.

Mary Cryan, 1st March