Randomness and Computation 2017/18
Week 11 tutorial sheet (12-1pm, Wednesday 4th April)

1. The tutorial for week 10 considered a 2×2 Markov chain for contingency tables, setting a worked example, and then the goal of proving *irreducibility*, *aperiodicity* and also, that the unique stationary distribution is uniform on the state space $\Sigma_{r,c}$.

   During the tutorial, it was only possible to cover part (a) and most of (b). Finish the remainder of this question.

2. Consider a process $X_0, X_1, X_2, \ldots$ with two states, 0 and 1. The process is governed by two matrices, $\mathbf{P}$ and $\mathbf{Q}$. If k is even, the values $P_{i,j}$ give the probability of going from state i to state j on the step from X_k to X_{k+1}. Likewise, if k is odd then the values $Q_{i,j}$ give the probability of going from state i to state j on the step from X_k to X_{k+1}. Explain why this process does not satisfy Definition 7.1 (from lecture 13) of a (time-homogeneous) Markov chain. Then give a process with a larger state space that is equivalent to this process and satisfies Definition 7.1.

   (this is exercise 7.3 from [MU])

3. Suppose that the 2-SAT Algorithm for finding a satisfying assignment (lecture 14) starts with an assignment chosen uniformly at random, rather than an arbitrary bad starting assignment. How does this affect the expected time until a satisfying assignment is found?

   (this is exercise 7.7 from [MU])

4. A colouring of a graph is an assignment of a colour to each of its vertices. A graph is k-colourable if there is a colouring of the graph with k colours such that no two adjacent vertices have the same colour (a *proper* colouring).

   Let G be a 3-colourable graph (this is all we know about G).

   (a) Show that there is *some* 2-colouring (not 3) of the *vertices* of G which satisfies the condition that no triangle of G is *monochromatic*.

   Hint: Start with the proper 3-colouring and turn it into a 2-colouring to satisfy the *no monochromatic triangles* property. This is short.

   (b) Now start with an arbitrary vertex 2-colouring of G. This is unlikely to be a proper colouring, and likely to even contain some monochromatic triangles. Consider the following algorithm:

   Start with the arbitrary 2-colouring (which might have monochromatic triangles).

   While there are still some monochromatic triangles in G, choose any such triangle and then flip the colour of a randomly chosen vertex (1/3 for each) of that triangle. Derive an upper bound on the expected number of such recolouring steps before the algorithm finds a 2-colouring where there are no monochromatic triangles.

   Hint: This has some similarities to our 2-SAT algorithm and analysis. When you think about the problem first (in terms of vertices having the "right" colour as in the target colouring) it may seem that there is a higher chance of moving to a colouring with a worse match than before; the trick is to exclude one of the colour classes when calculating how similar the current colouring is to the target (use (a) for inspiration).

This is based on 7.10 of [MU].

Mary Cryan, 28th March