Randomness and Computation 2018/19
Week 10 tutorial sheet (12-1pm, Tues 26th, Wed 27th March)

1. Suppose that the 2-SAT Algorithm for finding a satisfying assignment (lecture 15) 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])

2. 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].

3. In this question we consider Markov chains in general, and Markov chains for "knapsack" in particular.

   The knapsack problem is described by a collection of item sizes $a_1, \ldots, a_n$ (all satisfying $a_i > 0$) and a capacity $b > 0$. A feasible solution is any $\bar{x} \in \{0,1\}^n$ such that

$$\sum_{i=1}^{n} x_i a_i \leq b.$$

   Let Ω be the set of all feasible solutions. Our interest is in counting/estimating the number of different knapsacks (different $\bar{x} \in \{0,1\}^n$ tuples) which satisfy the condition above.

(a) Consider the following naïve method for estimating the number of knapsack solutions for a given input $a_1, \ldots, a_n$, $b > 0$: we generate $\bar{x} \in \{0,1\}^n$ uniformly at random a number of times (say k), and return 2^n, multiplied by the fraction of tests for which $\sum_{i=1}^n x_i a_i \leq b$ was satisfied.

Explain why we do not expect this approach to give a good estimate with high probability. It may help you to consider the situation where we have $b = \sqrt{n}$ and $a_i = 1$ for every i.

(b) Let us now define our "knapsack" Markov chain M, with Ω being the feasible vectors of $\{0,1\}^n$. Define the chain as follows:

$$M[\bar{x}, \bar{y}] = \begin{cases} 1/n & \exists j \in [n] \text{ such that } x_j = 1, y_j = 0, \text{ and } x_i = y_i \text{ for } i \neq j \\ 1/n & \exists j \in [n] \text{ such that } x_j = 0, y_j = 1, \text{ and } x_i = y_i \text{ for } i \neq j \text{ and} \\ & \sum_{i=1}^n a_i y_i \leq b \\ 0 & \text{for any } y \text{ that differs from } \bar{x} \text{ in 2 or more indices} \\ k/n & \text{for } \bar{y} = \bar{x}, \text{ if there are exactly } k \text{ } j\text{-indices such that} \\ & x_j = 0 \text{ but } (\sum_{i=1}^n a_i x_i) + a_j > b \end{cases}$$

This Markov chain models the process of choosing an index $j \in [n]$ uniformly at random, and then

- *either* changing x_j from 1 to become 0 in $\bar{y}$, keeping other positions identical (in this case $\bar{y}$s total size is definitely smaller than $\bar{x}$s)
- or if x_j instead was 0, we test whether $(\sum_{i=1}^n a_i x_i) \leq b - a_j$, and if so, we change x_j from 0 to become 1 in $\bar{y}$ (keeping the other positions identical). If the test fails, we set $\bar{y} = \bar{x}$.

Show that M has a stationary distribution, ie a row vector $\bar{\pi}$ (of length $|\Omega|$) satisfying $\bar{\pi} \cdot M = \bar{\pi}$, and that this stationary distribution is the uniform distribution, under the assumption that $\sum_{i=1}^n a_i > b$. You should show this in three steps:

i. Show that M is irreducible.

ii. Show that M is aperiodic.

iii. Demonstrate that the actual stationary distribution is the uniform one.

Mary Cryan, 24th March