Probability and Algorithms | **Caltech CS150, Fall 2014**
Leonard J. Schulman, schulman@caltech.edu | **TA: Nicholas Schiefer, nschiefer@caltech.edu**
Problem set 4 | **Out F 14/Nov Due W 26/Nov in class**

I will be traveling most of the week of Nov 17. Class will be held Monday morning (the 17th), but there will be no office hours that afternoon or class on Wednesday.

1. In this problem I'm going to ask you to prove a generalization of the threshold we studied for appearance of K_4 in $G(n,p)$. Given a graph H, let its "peak density" be

$$\rho(H) = \max_{H' \subseteq H} |E(H')|/|V(H')|$$

where $H' \subseteq H$ denotes that H' is a subgraph of H, and where V and E are the vertex and edge sets respectively.

You are to show that $p = n^{-1/\rho}$ is the threshold for appearance of H in $G(n,p)$. To simplify matters, it is enough for this problem if you show that for $\varepsilon > 0$, the probability tends to 0 for $p = n^{-1/\rho-\varepsilon}$ and to 1 for $p = n^{-1/\rho+\varepsilon}$.

Let $h = |V|$ and $e = |E|$. It is certainly possible for ρ to be larger than e/h. For example in the "flyswatter graph" (Fig. 1) with $h = 5, e = 6$, $\rho = 5/4$ and the threshold is $n^{-4/5}$ (rather than $n^{-5/6}$ as you might have been tempted to think).

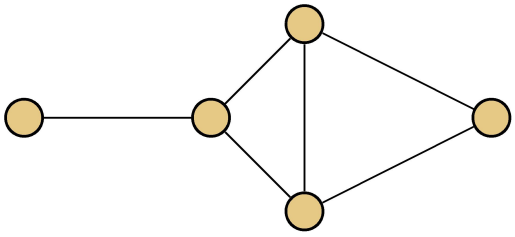

Figure 1: An abstract spatula

Hint: One of the superficial complications in this problem is that the automorphism group of H may be anything from trivial up through all permutations of the vertices (which is what we encountered for K_4). For instance the automorphism group of the spatula graph has size 2. This range of possibilities complicates counting how many ways H might turn up in the graph G that we sample from $G(n,p)$. The trick is that, since H is of fixed size, you can afford to count wastefully.

I want you to notice something carefully: there is in this problem a "fake threshold" $n^{-h/e}$. Once p rises above $n^{-h/e}$, the expected number of copies of X in G is $\omega(1)$. Despite this, with probability $1 - o(1)$ there are no copies of X in G until p reaches $n^{-1/\rho}$.

2. Consider the following modification of the hashing scheme described in class. As before we wish to hash n keys from the universe $[u]$ to an array of size $O(n)$. We are willing to tolerate $O(n)$ collisions, since those can be resolved by a secondary rehashing.

It is not too hard to find a pair of integers m, r so that $m \in \Theta(nr)$ is prime, and $m^{r-1} < u \leq m^{r+1}$. So suppose we have this pair. Now design a new hash family H' as follows. To each $\omega \in \mathbb{Z}/m$ assign a function $h_\omega : [u] \to \mathbb{Z}/m$ by

$$h_\omega(\kappa_0, \ldots, \kappa_r) = \sum \kappa_i \omega^i \bmod m.$$

Show that for any two keys $\kappa \neq \kappa'$, $\Pr_\omega(h_\omega(\kappa) = h_\omega(\kappa')) \in \Theta(1/n)$. Then show how this yields a hash family into an array of size $O(nr)$, so that you can quickly find a hash function with only $O(n)$ collisions, and in which a hash function requires only $O(\log n + \log\log u)$ bits to describe. (As compared with the FKS method described in class which requires $O(\log n + \log u)$ bits.)

Finally, show how to modify this into a collision-free hash family an array of size $O(n)$, in which only $O(\log n + \log\log u)$ bits need to be looked up in order to decide where to hash any key κ.

(This does not mean that the *total* amount of information required to describe the hash function is this small. For instance, in the FKS method, hashing a key κ requires looking up the $O(\log n + \log u)$ bits describing the main hash function $h_{h_0,\ldots,h_r}$, and then looking up another $O(\log n)$ to resolve collisions at the first hash destination. But the full description of the complete hash function requires $O(\log u + n \log n)$ bits. A similar contrast happens here.)

3. Show that the value of the ninefree problem, $f(n)$ (see previous problem set), is $O(n^{5/3})$.

   (In actual fact there is a matching $\Omega(n^{5/3})$ bound, but I am not assigning that.)