Joel Spencer

I hold a joint position in the Computer Science and the Mathematics Department. Detailed information about my research is available on my website, especially in the CV/Papers/Talks section.

My research is at the intersection of Probability and Discrete Mathematics, an area called Probabilistic Methods. This area was begun by the twentieth century Hungarian mathematician Paul Erdős. One examines random structures, with the goal of proving the existence of a combinatorial object. For example, in order to show the existence of a coloring with certain properties one examines the (appropriately defined) random coloring.

On more the CS side, one considers these random structures as objects *per se*. A key example is the random graph $G(n, p)$. This graph (technically, a probability distribution over graphs) consists of n vertices. Each pair of vertices is adjacent with (independent) probability p. As p increases this random graph *evolves* from empty to full. Certain areas merit special study. In particular, $p = \frac{1}{n}$ may be regarded as a *percolation point*. When p is appropriately smaller (say $\frac{0.99}{n}$) the graph consists of small components while when p is appropriately larger (say $\frac{1.01}{n}$) a "giant component" has emerged with a constant proportion of the vertices. These percolation events occur quite frequently in a wide variety of situations.

I've been interested in recent years in the study of *random processes*. One particular example is the study of the *random greedy algorithm*. Suppose, for example, one is given a collection of 3-element sets A_i, all subsets of a universe $\{1, \ldots, n\}$. One wants a "packing," a collection $A_{i_1}, \ldots, A_{i_s}$ of pairwise disjoint sets. Of course, $s \leq \frac{n}{3}$ and one wants to get as close to $\frac{n}{3}$ as possible. In the random greedy algorith, one permutes the A_i randomly and then considers them sequentially. When considered, A_i is added to the packing (which initially is empty) if possible – that is, unless some previous A_j had been placed in the packing which overlaps A_i. Analysis of this algorithm is surprisingly subtle, but one is able to show that under suitable side conditions it gives a "pretty good" (just how good remains a vexing open question!) packing.