Math 261B
Final take home
Due: 11am, Tuesday, March 20, 2012

1. A treap is a binary tree whose nodes contain two values, a *key* x, and a *priority* p_x. Keys are drawn from a totally ordered set and the priorities are given by a random permutation of the keys. Without loss of generality, we assume that the set of elements is $X = \{1, 2, \ldots, n\}$. The tree is a heap according to the priorities (i.e., if x is a parent of y, then $p_x < p_y$). And the tree is a search tree according to the keys, (i.e., if a node has a key x, then its left subtree contains nodes with keys $< x$ and its right subtree contains nodes with keys $> x$). For example, if $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $p = (3, 1, 5, 4, 8, 6, 2, 7)$, we have that $p_1 = 2, p_2 = 7, p_3 = 1$ and so on.

   Tweaps allow for fast insertion, deletion and search of an element. The cost of these operations is proportional to the height of the treap. In what follows you will show that this quantity is $O(\log n)$ with high probability. The analysis boils down to the following problems on random permutations on $[n] = \{1, 2, \ldots, n\}$: Given a permutation $p : [n] \to [n]$ of the n elements, an element is *checked* if it is larger than all the elements appearing to its left in p. For instance, if

$$p = (\mathbf{3}, 1, \mathbf{5}, 4, \mathbf{8}, 6, 2, 7), \tag{1}$$

   the elements that are checked are in bold. The problem is to show that the number of checks is concentrated around its expectation as described in the following subproblems:

   (1a) Given a key x, let x_- be the set of elements that are smaller than or equal to x, We will use p_-^x to denote the permutation induced by p on x_-. For example, using p from (1), we have that $p_-^6 = (3, 1, 5, 4, 6, 2)$. Show that all elements of x_- that are checked in p_-^x appear along the path from the root to x in the tree.

   (1b) Prove an analogous statement for the set x^+ of all elements $\geq x$ and use this to calculate exactly the number of elements from the root to x in the tree.

   (1c) Denoting with X_n the number of elements that are checked for a random permutation $p : [n] \to [n]$, prove that

$$E[X_n] = 1 + \frac{1}{2} + \ldots + \frac{1}{n}.$$

   (It is known that $H_n = \sum_{i=1}^{n} 1/i$, the nth *harmonic number*, is $\Theta(\log n)$.)

   (1d) Let Y_i be an indicator random variable denoting whether the ith element of the permutation (starting from the left) is checked. Prove that

$$\Pr[Y_i = 1 \mid Y_{i+1} = y_{i+1}, \ldots, Y_n = y_n] = \frac{1}{i}$$

for any choice of the y's.

(1e) Show that for any index set S,

$$\Pr\left[\bigwedge_{i\in S}(Y_i = 1)\right] \le \prod_{i\in S}\Pr[Y_i = 1]. \tag{2}$$

(1f) Prove that under the condition (2)the Chernoff bound holds for $Y = \sum_{i=1}^{n} Y_i$. Using this, give a concentration result for X_n.

2. The following type of geometric random graphs arises in the study of power control for wireless networks. We are given n points distributed uniformly at random within the unit square. Each point connects to the k-closest points. Let us denote the resulting (random) graph as G_k^n.

(2a) Show that there exists a contant α such that if $k \ge \alpha \log n$, then G_k^n is connected with probability at least $1 - 1/n$.

(2b) Show that there exists a constant β such that if $k \le \beta \log n$, then G_k^n is not connected with positive probability.

3. Consider the following parallel and distributed vertex-coloring algorithm. Every vertex u in a graph G initially has a list of colors $L_u = [\Delta(G) + 1]$ where $\Delta(G)$ denotes the maximum degree for vertices in G. The algorithm is in rounds. In each round the following happens.

- Every vertex not yet colored wakes up with probability 1/2.
- Every vertex that woke up picks a tentative color uniformly at random from its own list of colors.
- If t_u is the tentative color picked by u, and no neighbor of u picked t_u, then u colors itself with t_u.
- The color list of each uncolored vertex in the graph is updated by removing all colors successfully used by its neighbors.
- All uncolored vertices go back to sleep.

(3a) Show that the algorithm computes a legal coloring.

(3b) Show that in every round, each uncolored vertex colors itself with probability at least 1/4.

(3c) Show that within $O(\log |V(G)|)$ many rounds, the graph will be colored with high probability.

4. Let $f(x_1, \ldots, x_n)$ be a Lipschitz function with constant c. Namely, changing any coordinate changes the value of f by at most c. Let σ be a permutation of $[n]$ chosen uniformly at random. Show a strong concentration for $f(\sigma(1), \ldots, \sigma(n))$.

   Hint: Use a natural coupling to bound

$$\left| E[f \mid X_1, \ldots, X_{i-1}, X_i = a_i] - E[f \mid X_1, \ldots, X_{i-1}, X_i = b_i] \right|.$$