CPS130 - Homework 5
Due: Oct 27, 2005

1. Let X be a non-negative random variable such that E[X] is well defined. Using this, prove, for all $t > 0$, *Markov's Inequality:*

$$\Pr[X \geq t] \leq \frac{E[X]}{t}$$

2. One of the ways we can improve randomized Quicksort is to use a median of three partitioning scheme. With this method, we pull three randomly selected items out of the list, and we use the median of the three values as our pivot. Note that this removes the two worst cases for Quicksort, where the minimum or maximum is chosen to be the pivot. Assuming that there are at least three elements in the list and they all have distinct values.

   a) If we let x be the chosen pivot and $A'[1..n]$ be the sorted list, what is the probability we choose $x = A'[i]$? Denote this value p_i.
   b) How much have we increased the likely hood of choosing the median as the pivot element, as compared to the normal implementation? What is the limit as $n \to \infty$?
   c) If we define a "good" split to be a choice of pivot in the range $n/3 \leq i \leq 2n/3$, how much have we increased the likelihood of getting such a split? (*Consider approximating the sum with an integral*)
   d) Argue that this method does not change the $\Omega(n \log n)$ lower bound on Quicksort, only the constant factor involved.

3. Suppose that at each run of Quicksort, the splits are in the proportion $1 - \alpha$ to α, for some $0 < \alpha < 1/2$ is a constant. Show that the minimum depth of a leaf in the recursion tree is approximately $-\log n / \log \alpha$ and the maximum depth is approximately $-\log n / \log(1 - \alpha)$.

4. Show how we can use two stacks to implement a Queue such that the amortized cost of each *Enqueue* and *Dequeue* is O(1).

5. Suppose you are given a directed graph $G = (V, E)$ with a weight assigned to each *vertex*. Call this weight $w(v)$. In this graph, the directed arc (u, v) is present if and only if $w(u) \leq w(v)$. For example, there would be an arc from a vertex with weight 8 to vertices of weight 10 or 15, but not to vertices of weight 2, 3, or 6. Assume all weights are distinct, ie there are no vertices with identical weights.

   a) We call a graph *transitive* if for all vertices u, v, w, if there are two edges (u, v) and (v, w), then the edge (u, w) is present. Prove this graph is transient.
   b) Show that this graph has no cycles, meaning it's a DAG.
   c) Find an upper bound (Big-O notation) on the number of edges in the graph, if there are n vertices.
   d) Devise a sorting algorithm that is linear in the number of edges and give its asymptotic runtime.