1. Find an example of a random variable with a finite $j$th moment for $1 \leq j \leq k$ but an unbounded $k+1$st moment for some value of $k$.

2. A fixed point of a permutation $\pi : [n] \rightarrow [n]$ (where $[n]$ denotes the set $\{1, 2, \ldots, n\}$) is a value $j$ for which $\pi(j) = j$. Find the variance of the number of fixed points of a permutation chosen uniformly at random from all permutations. Use this to upper bound (with high probability) the number of fixed points in a randomly chosen permutation.

3. Given a black box that generates integers uniformly at random from $k$, we give an algorithm for constructing a random permutation of $[n]$. For each $i \in [n]$, for any $n \leq k$, we determine an $f(i)$ as follows: $f(1)$ is picked at random from $[k]$ using the black box. For $f(i)$, $i > 1$, pick a random number $r$ from $[k]$ using the black box. If $r$ is distinct from all $f(j)$, $j < i$, then $f(i) = r$; otherwise pick another random number from $[k]$. The random permutation is the numbers sorted in the order of increasing $f(i)$.

First show that this algorithm gives a permutation chosen uniformly at random from all permutations. What is the expected number of calls to the black box when $k = n$, and when $k = 2n$. Using a Chernoff bound, bound the probability of having to make more than $4n$ calls to the black box when $k = 2n$.

4. The randomized quicksort algorithm to sort a set $S$ of $n$ numbers is as follows:
• Pick a pivot i.e. an element $r$ of $S$ uniformly at random.
• Make a pass through $S$ and create sets $S_1 = \{x \in S : x \leq r\}$ and $S_2 = \{x \in S : x > r\}$.
• Recursively sort $S_1$ and $S_2$
• Output the sorted version of $S_1$ followed by $r$ followed by the sorted version of $S_2$.

Let us view the execution of Randomized quicksort as forming a tree where the root of each subtree is the pivot chosen for subset contained in that subtree. A node of this tree is called good if the pivot element in it divides the subtree rooted at that node into two sets each of size not more than $2/3$ of the subset contained in the subtree.

(a) Show that the expected running time of this algorithm is $O(n \log n)$.
(b) Show that the number of good nodes in any path from root to leaf in this tree is not greater than $c \log n$ for some constant $c$.
(c) Show that with probability at least $1 - 1/n^2$ the number of nodes in any given root to leaf path is not more than $c' \log n$ for some other constant $c'$.
(d) Show that, with high probability, the number of nodes in the longest root to leaf path is no more than $c' \log n$.
(e) Use these answers to show that the running time of Randomized quicksort is $O(n \log n)$ with high probability.