This exam lasts 90 minutes.
Write your answers in the separate answer booklet.
Please return this question sheet with your answers.

1. Assume we have access to a function RANDOM(k) that returns, given any positive integer k, an integer chosen independently and uniformly at random from the set $\{1, 2, \ldots, k\}$, in $O(1)$ time. For example, to perform a fair coin flip, we could call RANDOM(2).

   Now suppose we want to write an efficient function RANDOMPERMUTATION(n) that returns a permutation of the set $\{1, 2, \ldots, n\}$ chosen uniformly at random; that is, each permutation must be chosen with probability $1/n!$.

   (a) ***Prove*** that the following algorithm is ***not*** correct. *[Hint: Consider the case $n = 3$.]*

   ```
   RandomPermutation(n):
     for i ← 1 to n
         π[i] ← i
     for i ← 1 to n
         swap π[i] ↔ π[Random(n)]
     return π
   ```

   (b) Describe and analyze a correct RANDOMPERMUTATION algorithm that runs in $O(n)$ expected time. (In fact, $O(n)$ worst-case time is possible.)

2. Suppose we have n pieces of candy with weights $W[1..n]$ (in ounces) that we want to load into boxes. Our goal is to load the candy into as many boxes as possible, so that each box contains at least L ounces of candy. Describe an efficient 2-approximation algorithm for this problem. ***Prove*** that the approximation ratio of your algorithm is 2.

   (For 7 points partial credit, assume that every piece of candy weighs less than L ounces.)

3. The MAXIMUM-k-CUT problem is defined as follows. We are given a graph G with weighted edges and an integer k. Our goal is to partition the vertices of G into k subsets $S_1, S_2, \ldots, S_k$, so that the sum of the weights of the edges that cross the partition (that is, with endpoints in different subsets) is as large as possible.

   (a) Describe an efficient randomized approximation algorithm for MAXIMUM-k-CUT, and ***prove*** that its expected approximation ratio is ~~at most~~ $(k-1)/k$.

   (b) Now suppose we want to minimize the sum of the weights of edges that do *not* cross the partition. What expected approximation ratio does your algorithm from part (a) achieve for this new problem? ***Prove*** your answer is correct.

4. The citizens of Binaria use coins whose values are powers of two. That is, for any non-negative integer k, there are Binarian coins with value is 2^k bits. Consider the natural greedy algorithm to make x bits in change: If $x > 0$, use one coin with the largest denomination $d \le x$ and then recursively make $x - d$ bits in change. (Assume you have an unlimited supply of each denomination.)

   (a) ***Prove*** that this algorithm uses at most one coin of each denomination.

   (b) ***Prove*** that this algorithm finds the minimum number of coins whose total value is x.

5. Any permutation π can be represented as a set of disjoint cycles, by considering the directed graph whose vertices are the integers between 1 and n and whose edges are $i \rightarrow \pi(i)$ for each i. For example, the permutation $\langle 5,4,2,6,7,8,1,3,9 \rangle$ has three cycles: $(175)(24683)(9)$.

   In the following questions, let π be a permutation of $\{1, 2, \ldots, n\}$ chosen uniformly at random, and let k be an arbitrary integer such that $1 \le k \le n$.

   (a) ***Prove*** that the probability that the number 1 lies in a cycle of length k in π is precisely $1/n$. *[Hint: Consider the cases $k = 1$ and $k = 2$.]*

   (b) What is the *exact* expected length of the cycle in π that contains the number 1?

   (c) What is the *exact* expected number of cycles of length k in π?

   (d) What is the *exact* expected number of cycles in π?

   You may assume part (a) in your solutions to parts (b), (c), and (d).