CIS507: Design & Analysis of Algorithms
Tutorial: Sorting & Probabilistic Analysis

1 Indicator Random Variables

1. Let X be an indicator random variable such that $E[X] = 1/2$. What is $E[\sqrt{X}]$?

2. Let X be an indicator random variable. Which of the following statements are true? (Circle all that apply)

   (a) $Pr\{X = 0\} = Pr\{X = 1\} = 1/2$

   (b) $Pr\{X = 1\} = E[X]$

   (c) $E[X] = E[X^2]$

   (d) $E[X] = \big(E[X]\big)^2$

2 Throwing into Buckets

Let $A[1, \ldots, n]$ be an array of numbers uniformly distributed over the interval $[0, 1)$. Suppose we have another array $B[0, n-1]$ of buckets (implemented as lists). Consider the following loop, after which bucket i holds values of A in the half-open interval $[i/n, (i+1)/n)$:

for i = 1 **to** n **do**
  insert $A[i]$ into bucket $B[\lfloor nA[i] \rfloor]$

1. Trace the algorithm on $A = [.78, .17, .39, .26, .72, .94, .21, .12, .23, .68]$. Show the contents of array B.

2. Define random variable n_i = the number of elements placed in bucket $B[i]$ (note that this is *not* an indicator random variable). Prove that $E[n_i] = 1$ for $i = 0, 1, \ldots, n-1$. That is, the expected number of items in each bucket is 1.

   Hint: $\forall i = 0, 1, \ldots, n-1$ and $j = 1, \ldots, n$, define indicator random variables:

   $$X_{ij} = I\{A[j] \text{ falls in bucket } i\} = \begin{cases} 1 & \text{if } A[j] \text{ falls in bucket } i \\ 0 & \text{otherwise} \end{cases}$$

   Start by defining n_i in terms of the X_{ij}'s.

3. Prove that $E[n_i^2] = 2 - \frac{1}{n}$ for $i = 0, 1, \ldots, n-1$.

   Hint: Note that X_{ij} and X_{ik} are independent for $j \neq k$.

3 Bucket Sort

Bucket sort assumes the input is generated by a random process that distributes elements uniformly over $[0, 1)$. It divides $[0, 1)$ into n equal sized buckets, distributed the n inputs values into the buckets, sorts each bucket (with insertion sort), then concatenates all buckets. The pseudo code is shown below.

```
BUCKET-SORT(A, n)
  let B[0 .. n − 1] be a new array
  for i = 0 to n − 1
      make B[i] an empty list
  for i = 1 to n
      insert A[i] into list B[⌊n · A[i]⌋]
  for i = 0 to n − 1
      sort list B[i] with insertion sort
  concatenate lists B[0], B[1], ..., B[n − 1] together in order
  return the concatenated lists
```

Intuitively, if each bucket gets a constant number of elements, it should take $\mathcal{O}(1)$ time to sort each bucket, therefore $\mathcal{O}(n)$ to sort all buckets.

1. Define random variable n_i = the number of elements placed in bucket $B[i]$. Write the recurrence of the above algorithm.

2. Show that the expected running time of Bucket Sort is $E[T(n)] = \Theta(n)$.

Hints:

- Remember that "insert sort" is $\Theta(n^2)$.
- Use the fact that $E\left[n_i^2\right] = 2 - \frac{1}{n}$, which we proved in the previous question.
- If X is a random variable, and a is a constant, then $E[aX] = aE[X]$. Consequently, $E[\mathcal{O}(g(n))] = \mathcal{O}(E[g(n)])$, since the expectation doesn't depend on lower-order terms.