Practice Problem Set 6: Continuous Random Variables

Reading: The textbooks do not cover the material, thus your class notes are the definite reference. Chapter 8 in MU provides a good introduction to continuous random variables and distributions, but it does not cover all the material we saw in class.

Exercise 1. Prove that the two definitions of independence we saw are equivalent for discrete random variables.

Definition 1: Two discrete random variables X and Y are independent if, for all values $x \in \text{Range}(X)$ and $y \in \text{Range}(Y)$, the events $[X = x]$ and $[Y = y]$ are independent.

Definition 2: Two discrete random variables X and Y are independent if, for all values $x \in \mathbb{R}$ and $y \in \mathbb{R}$, the events $[X \leq x]$ and $[Y \leq y]$ are independent.

Exercise 2. Alice wants to sell her Pokemon cards collection on Ebay, and she decided to accept the first bid that exceeds K dollars. Assuming that bids are independent continuous random variables with the same CDF F, find the expected number of bids Alice receives before she sells the Pokemon collection.

Exercise 3. Let X and Y be independent continuous random variables with the same CDF F and PDF f. Show that $V = \max\{X, Y\}$ has CDF $\Pr(V \leq x) = F(x)^2$ and PDF $f_V(x) = 2f(x)F(x)$ for all $x \in \mathbb{R}$. Find the CDF of $U = \min\{X, Y\}$.

Exercise 4. In ProbabilityVille, the amounts of rain that fall each year are independent, identically distributed, random variables $\{X_i : i \geq 1\}$. Find the probability that:

(*a*) $X_1 < X_2$.

(*b*) $X_1 > X_2 > X_3 > X_4$.

(*c*) $X_1 < X_2$ and $X_2 > X_3$ and $X_3 < X_4$.

For this final exercise, we go back to the discrete setting to see another example of a randomized algorithm.

Exercise 5. (Data stream sampling) A data stream is an extremely long sequence of items that you can only read only once, in order. A good example of a data stream is the sequence of packets that pass through a router. Data stream algorithms must process each item in the stream quickly, using very little memory; there is simply too much data to store, and it arrives too quickly for any complex computations. Every data stream algorithm looks roughly like this:

```
DoSomethingInteresting(Stream S)
  repeat:
      x ← next item in S
      ⟨⟨do something interesting with x⟩⟩
  until S ends
  return ⟨⟨something⟩⟩
```

Describe an algorithm that chooses one element uniformly at random from a data stream, *without knowing the length of the stream in advance*. Prove that your algorithm is correct. Your algorithm should spend $O(1)$ time per stream element and use $O(1)$ space (not counting the stream itself).