Randomized Algorithms 2023
Solutions to Tutorial Sheet 3

1. (a) By Definition,

$$\begin{aligned}
\mathrm{Var}[X-Y] &= \mathrm{E}[(X-Y)^2] - \mathrm{E}[X-Y]^2 \\
&= \mathrm{E}[X^2] - 2\mathrm{E}[XY] + \mathrm{E}[Y^2] - (\mathrm{E}[X]^2 - 2\mathrm{E}[X]\mathrm{E}[Y] + \mathrm{E}[Y]^2) \qquad \text{(by linearity)} \\
&= \mathrm{E}[X^2] - 2\mathrm{E}[X]\mathrm{E}[Y] + \mathrm{E}[Y^2] - \mathrm{E}[X]^2 + 2\mathrm{E}[X]\mathrm{E}[Y] - \mathrm{E}[Y]^2 \\
& \qquad\qquad \text{(by independence)} \\
&= \mathrm{Var}[X] + \mathrm{Var}[Y].
\end{aligned}$$

   (b) We use conditional expectation:

$$\begin{aligned}
\mathrm{E}[X^3] &= \mathrm{E}[X^3 \mid X = 1]\Pr[X = 1] + \mathrm{E}[X^3 \mid X > 1]\Pr[X > 1] \\
&= p + \mathrm{E}[(X+1)^3](1-p) \\
&= p + (\mathrm{E}[X^3] + 3\mathrm{E}[X^2] + 3\mathrm{E}[X] + 1)(1-p),
\end{aligned}$$

   which simplifies into

$$p\mathrm{E}[X^3] = 1 + 3\mathrm{E}[X^2](1-p) + 3\mathrm{E}[X](1-p).$$

   We have known that for a geometric random variable $\mathrm{E}[X] = 1/p$ and $\mathrm{E}[X^2] = \frac{2-p}{p^2}$. Thus,

$$\mathrm{E}[X^3] = \frac{p^2 - 6p + 6}{p^3}.$$

2. (Rough solution to second Q) Imagine that we draw the n inputs to BucketSort independently and uniformly at random from $\{0,1\}^k$. Hence ...

   The first-m-bits of the inputs are independently uniform from $\{0,1\}^m$. Each a_i has probability $\frac{1}{2^m}$ of entering any bucket. Bucket Sort can be seen as a "balls-in-bins" experiment.

   The running time of step (a.) of the algorithm is $\Theta(n + 2^m)$, which is time for both the linear scan of the inputs, after initializing the array of length 2^m buckets. The *expected* running time for the for loop in steps (b.) and (c.) will be $\mathrm{E}[\sum_{b\in\{0,1\}^m} c \cdot (X_b^2)]$, where X_b is the number of inputs landing in bucket b, and $c > 0$ is the fixed constant of the $O(n^2)$ algorithm.

   We want to evaluate $\mathrm{E}[\sum_{b\in\{0,1\}^m} c \cdot (X_b^2)] = \sum_{b\in\{0,1\}^m} c \cdot \mathrm{E}[X_b^2]$.

   We are now going to use an unexpected "trick" where we exploit the "second moment" of Binomial random variables to bound the $\mathrm{E}[X_b^2]$.

   Realise each X_b is a binomial random variable $B[n, \frac{1}{2^m}]$ with

$$\mathrm{E}[X_b^2] = n(n-1)2^{-2m} + n2^{-m}.$$

Multiplying by 2^m (for each $b \in \{0,1\}^m$), and by c, this gives expected time for (b.)-(c.) at most

$$c \cdot \left(n^2 2^{-m} + n \right).$$

We choose m carefully, letting $m = \lceil \lg(n) \rceil$. We see that this ensures that the time taken for the scan and intialization is $\Theta(n + 2^m) = \Theta(n)$, and the expected number of steps for (b)-(c) is at most $2 \cdot c \cdot n$ which is $\Theta(n)$, so the overall running time is indeed $\Theta(n)$.

3. Consider a function $F : \{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, m-1\}$ and suppose we know that for $0 \leq x, y \leq n-1$, $F((x+y) \bmod n) = (F(x) + F(y)) \bmod m$. The only way we know to evaluate $F(\cdot)$ is to examine the values in an array where the $F(\cdot)$ values have been stored (with entry i holding the value of $F(i)$). Unfortunately, a system failure has corrupted up to a $1/5$-fraction of the entries of the array, so we no longer have reliable values in all positions. We now design a simple randomized algorithm that, given an input $z \in \{0, \ldots, n-1\}$, outputs a value that equals $F(z)$ with probability at least $1/2$.

   note: we are not allowed to assume that the corruptions have taken place uniformly at random; it is possible that they could have ocurred along a contiguous sub-block of the array, or maybe just at even positions of the array. So we must take a worst-case approach to the analysis.

   solution: Our algorithm relies on the additive (modulo m) property and a simple sampling rule: regardless of the value z input, we will randomly draw a value y from $\{0, 1, \ldots, n-1\}$ uniformly at random. After having drawn y, we will define $x =_{\text{def}} (z - y) \bmod n$. Note that this ensures $z = (x + y) \bmod n$. We will then "lookup" the values of $F(x)$ and $F(y)$ from the table, add $F(x)$ and $F(y)$ together, and take the remainder with m as the final answer for $F(z)$.

   The probability that this value returned is the true value $F(z)$ is exactly the probability that *both* $F(x)$ and $F(y)$ were *not corrupted*. Let's consider the fact that at most $\frac{n}{5}$ of the n entries of F have been corrupted; this means that in considering the n different (x, y) *pairs*, at most $\frac{2n}{5}$ of these might be somehow corrupted (either x corrupted, or y corrupted, or both). So $\frac{3n}{5}$ of the n (x, y) pairs for z are uncorrupted, and given that we drew y randomly from all possible n values, this means the probability we have an uncorrupted pair is $\geq \frac{3}{5} > \frac{1}{2}$. This result is independent of the particular z which was chosen.

   Now suppose we are allowed to repeat the initial algorithm three times before we return a result. When it says "repeat the algorithm" this implies a resampling of the y (and hence a new (x, y) pair) each time. The approach should be to look at the three results returned for $F(z)$ and if ≥ 2 of those are the same, return that value; otherwise, if all are different, randomly return any of them.

   What is the probability of a correct answer? It is certainly at least as high as the probability that *at most one answer is corrupted*, which is (just using, from the binomial distribution, the sum of the probability of 3 successes, plus that of 2 successes, in 3 bernoulli trials, with probability of success 3/5 in each trial), given by $\frac{3}{5}^3 + 3\frac{2}{5}\frac{3}{5}^2 = 0.648$.

   Note that this is greater than $\frac{3}{5}$. Hence, by using three trials in this way, we can improve the probability of giving the correct answer.

Note that if exactly two of the results are corrupted, which happens with probability $3\frac{3}{5}\frac{2}{5}^2 = 0.432$, it is still possible that the right answer will be returned, in particular if the two corrupted answers are unequal, in which case one of the three answers will be chosen at random, and with $\frac{1}{3}$ probability the right answer will be returned. Unfortunately, it is not guaranteed that the two corrupted answers will be unequal, and we may be tricked into giving the wrong answer if the two corrupted values match. Indeed, we have no control over corrupted answers, and all corrupted answers may be equal. So we can only say that the total probability of the right answer being returned in the scheme with 3 trials is at least .648, which is an improvement on the original 3/5 probability with one trial.