Randomized Algorithms—Problem Set #1—Fall 2022.

1. Intuition in gambling is a dangerous thing. We meet a swindler who offers us the following game. There are three cards: one shows two spades, one on each side; the second card has a diamond on both sides; the third has a spade in one side and a diamond on the other. The swindler shakes the three cards in his hat, picks one and puts it on the table. He bets the card will show the same suit as the one visible at the top. Thus, for example, if the chosen card shows a diamond on top, he bets \$1 that the other side is also a diamond. He says "if the card is diamond-diamond, you pay me \$1; if it is diamond-spade then I pay you \$1. The card can't be spade-spade, so there is equal chance that any of us wins, it's a fair game.". After some rounds you're losing some money, so in order to keep playing he says you must have been specially unlucky an he offers to pay \$1.25 if you win, whereas you'll only have to pay \$1 when you lose. However, after a few more rounds the swindler has plucked you.

   (a) What's the catch? If the (first) game were fair your expected gain (and well as his) per round would be 0. Show that the game is not fair. What is your expected gain (actually, loss) if you play 100 rounds of the original game?

   (b) What is your expected gain in the second variant? Show that it is not $-1/2 + 1/2 \times 1.25 = -0.5 + 0.625 = \$0.125$. What is your expected total gain (loss) if you play 100 rounds of this second game?

2. You roll a standard die $N = 10$ times. What is the probability that the sum of the N rolls is a multiple of 6? What if you throw the die $N = 20$? Hint: use the principle of deferred decisions.

3. Given $X \sim \text{Geom}(p)$ compute $\mathbb{P}[X > k]$.

4. Show that if $X_1 \sim \text{Poisson}(\lambda_1)$ and $X_2 \sim \text{Poisson}(\lambda_2)$ are independent random variables then $X_1 + X_2 \sim \text{Poisson}(\lambda_1 + \lambda_2)$.

5. Median-of-$(2t + 1)$ quickselect is a variant of quickselect which picks uniformly at random and without replacement a sample of $2t + 1$ elements from the array and uses the median of these $2t + 1$ elements as the pivot for each recursive stage (as long as $n \geq 2t + 1$). So the probability $\pi_{n,k}$ that the pivot choosen is the k-th smallest element is not uniform, while

$$\pi_{n,k} = \frac{1}{n}, \qquad \text{for all } k,\ 1 \leq k \leq n,$$

   in ordinary quickselect, when the pivot of each stage is choosen u.a.r.

   When analyzing the expected cost of ordinary quickselect we set up a recurrence for the expected number $f_n = \mathbb{E}[F_n]$ of comparisons to select an element of random rank as follows:

$$f_1 = f_0 = 0$$

$$f_n = n - 1 + \sum_{k=1}^{n} \pi_{n,k} \times \mathbb{E}[\text{\# of comparisons} \,|\, \text{pivot is the } k\text{-th element}]$$

$$= n - 1 + \sum_{k=1}^{n} \pi_{n,k} \left(\frac{k-1}{n} f_{k-1} + \frac{n-k}{n} f_{n-k} \right) = n - 1 + \frac{2}{n^2} \sum_{k=0}^{n-1} k f_k.$$

Similar steps can be applied in the case of median-of-$(2t+1)$ quickselect, but the so called *splitting probabilities* $\pi_{n,k}$ are different. Also the number of comparisons of each recursive stage will be larger because we need some comparisons to find the median of the sample. However, t is a constant and hence the number of comparisons in a recursive stage of the algorithm is $n + \mathcal{O}(1)$.

(a) Calculate the probability $\pi_{n,k}$ that the pivot is the k-th smallest element, when it is choosen as the median of $2t+1$ random elements selected without replacement from the array of n elements.

(b) Set up the recurrence for the expected number $f_n^{(t)}$ of comparisons to select an element of random rank for quickselect with median-of-$(2t+1)$.

(c) Identify a shape function for the recurrence. Apply the continuous master theorem to solve the recurrence. Show that $f_n^{(t)} = \frac{2t+3}{t+1} n + o(n)$ (for ordinary quickselect we have $f_n = f_n^{(0)} = 3n + o(n)$).

Useful formulas:

- If k is constant with respect to x and $x \to \infty$ then

$$\binom{x}{k} \sim \frac{x^k}{k!}$$

- (Beta integral) For any $m, n \geq 0$,

$$\int_0^1 x^m (1-x)^n \, dx = \frac{m!n!}{(m+n+1)!}$$

6. Modify the Randomized Selection algorithm so that it can be used to select the j-th smallest element (the algorithm presented in class is only to select the median). You will only need to carefully choose d and u so that the j-th smallest element of the input array S belongs to C with high probability. You'll need a few other minor adjustements, e.g., in step 6, the algorithm fails if $l_d > j$ or $l_u > n - j$.

(a) Prove that the algorithm will either correctly report the j-th smallest element in S or fail.

(b) Check that the modified algorithm still has cost $\Theta(n)$.

(c) Prove that the probability of failure is $\leq n^{-1/4}$.

(d) Suppose that in order to extract C in step 5, you compare first $S[i]$ to d if $j < \lceil n/2 \rceil$ and then to u only if $S[i] \geq d$ (for otherwise $S[i] \notin C$). Likewise, if $j \geq \lceil n/2 \rceil$ compare first $S[i]$ to u and then compare to d only $S[i] \leq u$. The number of comparisons in step 5 will always be between n and $2n$, but what is the average number of comparisons (as a function of j)?