Collaboration Statement: Andrew Koulogeorge and Eric Richardson

Problem 2: Another Exercise in Union Bound.

Solution. We know that our randomized algorithm A has the following property for any random bit vector z:

$$\forall \mathbf{z} \in \{0,1\}^n, \quad \underset{\mathbf{r}\in\{0,1\}^n}{\mathbf{Pr}}[\mathbf{z} \text{ fools } \mathbf{r}] \leq \frac{1}{9}$$

We wish find the smallest value for T, the number of random bit vectors in the set R such that the probability that *any* bit vector z fools R is at most 1. Let T be the number of random bit vectors in R. By definition, the probability that **z** fools R is equal to the probability that **z** fools at least half the vectors in R.

Let X be a random variable such that X equals the number of random vectors $r \in R$ that z fools for an arbitrary z. Note from above that the probability that z fools any random bit vector r is at most $\frac{1}{9}$. Since each of the random vectors are drawn independently from each other, I claim that X is a binomial random variable with the T independent trails being a success if z fools r_i and a failure otherwise. More compactly, $X \sim \text{Bin}(T, \frac{1}{9})$. Thus, $\mathbb{P}[z \; fools \; R] = \mathbb{P}\left[X \geq \frac{T}{2}\right]$

$$\mathbb{P}\left[X \geq \frac{T}{2}\right] = \sum_{k=\lceil T/2 \rceil}^{T} \binom{T}{k} \left(\frac{1}{9}\right)^k \left(\frac{8}{9}\right)^{T-k}$$

First, observe that the probability term $\left(\frac{1}{9}\right)^k \left(\frac{8}{9}\right)^{T-k}$ is monotonically decreasing as k goes from $0 \to T$. This implies that the largest value of $\left(\frac{1}{9}\right)^k \left(\frac{8}{9}\right)^{T-k}$ in the above summation is when $k = \lceil T/2 \rceil$:

$$\mathbb{P}\left[X \geq \frac{T}{2}\right] \leq \sum_{k=\lceil T/2 \rceil}^{T} \binom{T}{k} \left(\frac{1}{9}\right)^{\lceil T/2 \rceil} \left(\frac{8}{9}\right)^{\lceil T/2 \rceil}$$

Second, observe we can upper bound the number of terms in the above sum by noting that there are 2^T total subsets of R since $|R| = T$:

$$\mathbb{P}\left[X \geq \frac{T}{2}\right] \leq 2^T \left(\frac{1}{9}\right)^{\lceil T/2 \rceil} \left(\frac{8}{9}\right)^{\lceil T/2 \rceil}$$

By rewriting exponents and noting that since $\frac{8}{9} < 1$, removing it from the term makes the overall value larger:

$$\mathbb{P}\left[X \geq \frac{T}{2}\right] \leq 2^T \left(\frac{1}{3}\right)^T = \left(\frac{2}{3}\right)^T$$

Now that we have bounded the probability that z fools the set of bit vectors R, lets consider the probability that there exists a z which fools R. Let I be the index set containing all possible binary vectors z. Note that $|I| = 2^n$

$$\mathbb{P}\left[\exists\ z\ s.t\ z\ fools\ R\right] = \mathbb{P}\left[\bigcup_{i \in I} z_i\ fools\ R\right]$$

$$\mathbb{P}\left[\exists\ z\ s.t\ z\ fools\ R\right] \leq \sum_{i=1}^{2^n} \mathbb{P}\left[z_i\ fools\ R\right] \leq 2^n \left(\frac{2}{3}\right)^T$$

where the above implication is a result of the union bound. Note that now we are reasoning over all possible binary vectors z but are random variable above applies for an arbitrary z. We can not solve for T when bounding the above probability by 1:

$$2^n \left(\frac{2}{3}\right)^T < 1 \implies \log 2^n + T \log \frac{2}{3} < \log 1 \implies n + T(1 - \log_2 3) < 0 \implies T = O(n) \quad \square$$

Solution.

We are given the following:

$$\forall \mathbf{z} \in \{0,1\}^n, \quad \underset{\mathbf{r} \in \{0,1\}^n}{\mathbf{Pr}}[\mathbf{z} \text{ fools } \mathbf{r}] \leq \frac{1}{9}$$

Given a set of random bit strings $R = \{r^{(1)}, r^{(2)}, ..., r^{(T)}\}$ with each $r^{(t)} \in \{0,1\}^n$, we wish to determine

$$\mathbf{Pr}[\mathbf{z} \text{ fools at least half of R}]$$

Let $A = \{R_A \subseteq R : |R_A| \geq \frac{T}{2}\}$. Then, for any **z**, the above can be stated equivalently as:

$$\mathbf{Pr}[\exists R_A \in A : \mathbf{z} \text{ entirely fools } R_A]$$

where "**z** entirely fools R_A" means **z** fools *every* string in R_A. It follows that

$$\begin{aligned}
\textbf{Pr}[\exists R_A \in A : \textbf{z} \text{ entirely fools } R_A] &= \textbf{Pr}[\bigcup_{R_A \in A} \textbf{z} \text{ entirely fools } R_A] && (1)\\
&\leq \sum_{R_A \in A} \textbf{Pr}[\textbf{z} \text{ entirely fools } R_A] && (2)\\
&\leq \sum_{R_A \in A} \frac{1}{9^{\frac{T}{2}}} && (3)\\
&\leq \sum_{R_A \in A} \frac{1}{3^T} && (4)\\
&\leq \frac{2^T}{3^T} && (5)
\end{aligned}$$

We have (2) by the union-bound. (3) follows from the fact that all $R_A \in A$ contain at least $\frac{T}{2}$ bit strings. This implies that **Pr[z** entirely fools R_A**]** can be no greater than the probability that it fools exactly $\frac{T}{2}$ individual bit strings, which is $\frac{1}{9^{\frac{T}{2}}}$. Because A is simply a collection of subsets of R, $|A|$ can be no greater than 2^T, giving us (5). Hence, **Pr[z** fools at least half of R**]** $\leq \frac{2^T}{3^T}$. Consider all 2^n possible bit strings $\textbf{z} \in \{0,1\}^n$. We now determine the probability that any $\textbf{z} \in \{0,1\}^n$ fools R. Applying the union bound again, we have

$$\begin{aligned}
\textbf{Pr}[\text{any } \textbf{z} \text{ fools } R] &= \textbf{Pr}[\bigcup_{\textbf{z} \in \{0,1\}^n} \textbf{z} \text{ fools at least half of } R]\\
&\leq \sum_{\textbf{z} \in \{0,1\}^n} \textbf{Pr}[\textbf{z} \text{ fools at least half of } R]\\
&\leq \sum_{\textbf{z} \in \{0,1\}^n} \frac{2^T}{3^T}\\
&= \frac{2^n 2^T}{3^T}
\end{aligned}$$

The task now becomes fixing a T such that this quantity is less than 1:

$$\begin{aligned}
\frac{2^n 2^T}{3^T} &< 1\\
2^n 2^T &< 3^T\\
2^{n+T} &< 3^T\\
2^{n+T} &< 2^{T\log_2 3}\\
n + T &< T\log_2 3\\
T\log_2 3 - T &> n\\
T &= O(n)
\end{aligned}$$

Therefore, to ensure that the probability an input **z** fools R is less than 1, we should choose a T of size $O(n)$.

Problem 5: Checking Matrix Multiplication.

Solution.

procedure CHECKMATMUL($n \times n$ matrices A, B, and C)
  $r \leftarrow n$ random bits from $\{0, 1\}$
  $D_r \leftarrow A(Br)$
  $C_r \leftarrow Cr$
  return $D_r = C_r$

- **Runtime**:
  Observe that r is a bit vector of size n. As a result, the matrix-vector product Br is a vector of size n, where the ith entry in Br is given by $\sum_{k=1}^{n} B_{ik} r_k$. This operation requires a summation over n for each $i \in [n]$, resulting in a total of n^2 operations. By the same logic, $A(Br)$ and Cr require n^2 additions. This results in two vectors $D_r = A(Br)$ and $C_r = Cr$, both of length n. Thus, the final equality check between D_r and C_r requires n comparisons. Let $T(n)$ denote the number of operations made by CHECKMATMUL(A, B, C) with each A, B, C of dimensions $n \times n$. It follows that $T(n) = 3n^2 + n = O(n^2)$.

- **Correctness**:
  First, consider the case where $AB = C$. Here, we are effectively computing whether $Cr = Cr$, which is clearly true for any r. Hence, our algorithm has the property that **Pr**[CHECKMATMUL(A, B, C) is correct $| \, AB = C] = 1$. In other words, our algorithm has no false negatives.

  Now, lets consider the case when $AB \neq C$. We want to bound the probability that our algorithm makes an error. That is, we want to bound the probability that CHECKMATMUL(A,B,C) returns True given $AB \neq C$. If we can bound the probability of this event, because we have shown that our algorithm has no false negatives, we can then apply boosting to achieve an error of any $\delta > 0$

$$ABr = Cr \implies (AB - C)r = 0$$

  Following the method used by DeepC in class in the equality checking problem, let $Z = AB - C$. Thus, we are interested in the probability of

$$\mathbb{P}\left[Zr = 0\right] = \mathbb{P}\left[\bigcap_{i=1}^{n} z^i \cdot x = 0\right]$$

where z^i represents the rows of the matrix Z. By assumption, $Z \neq 0$ since $AB \neq C$. Since Z is not the zero matrix $\exists$ a row vector z^k of Z such that $z^k \neq 0$. Let us now consider only the probability of this row vector $z^k \cdot x = 0$ instead of all row vectors in Z:

$$\mathbb{P}[Zr = 0] = \mathbb{P}\left[\bigcap_{i=1}^{n} z^i \cdot x = 0\right] \leq \mathbb{P}\left[z^k \cdot x = 0\right]$$

Let us define the above event as A. We then want to understand:

$$\mathbb{P}[A] = \mathbb{P}\left[\sum_{i=1}^{n} z_i^k x_i = 0\right]$$

We proceed by following the argument that DeepC made in class. Since this row vector z^k is nonzero, we can say wlog that $z_1 \neq 0$. Let us now define $X = z_1 r_1$ and $Y = \sum_{i=2}^{n} z_i^k x_i$. By applying the law of total probability and conditioning on the values of the random variable Y, we see that:

$$\mathbb{P}[A] = \mathbb{P}[A|Y = 0]\,\mathbb{P}[Y = 0] + \mathbb{P}[A|Y \neq 0]\,\mathbb{P}[Y \neq 0]$$

First, observe that if we condition on $Y = 0$ then A can only occur when $z_1 r_1 = 0$ which occurs if and only if $r_1 = 0$, which occurs with probability $\frac{1}{2}$, since we assumed that $z_1 \neq 0$. Second, recall that given an event X, $\mathbb{P}[X] + \mathbb{P}\left[\overline{X}\right] = 1$. We can apply this to the two events where $Y = 0$ and $Y \neq 0$ and define $\mathbb{P}[Y = 0] = q$ and $\mathbb{P}[Y \neq 0] = 1 - q$. Third, note that if we condition on $Y \neq 0$ then A can only occur when $z_1 r_1 = -Y$. Note here the important distinction about what is random in this problem and what isn't. z_1 is NOT a random variable; indeed, it is an arbitrary and fixed element which could be chosen by our worst adversary. r_1, however, is random. Therefore, we must consider the various cases on the values in which z_1 can take on:

- If $z_1 \neq -Y$, then $\mathbb{P}[A|Y \neq 0] = 0$ since no value of r_1 will make $z_1 r_1 = -Y$
- If $z_1 = -Y$, then $\mathbb{P}[A|Y \neq 0] = \frac{1}{2}$ since $z_1 r_1 = -Y \iff r_1 = 1$ which occurs with probability = $\frac{1}{2}$

Thus, for any input we can say that $\mathbb{P}[A|Y \neq 0] \leq \frac{1}{2}$. Putting these 3 arguments together, we see that:

$$\mathbb{P}[Error] = \mathbb{P}[Zr = 0] = \mathbb{P}[A] \leq \frac{1}{2}q + \frac{1}{2}(1 - q) \leq \frac{1}{2}$$

Now that we have shown that we have bounded the probability of error for false positives and have shown our algorithm makes no false negative errors, we can apply boosting to our algorithm. We sample k random vectors and check $ABx =$

Cx for each of the k random vectors x. If all of the k of the resulting vectors are equal then our algorithm will return True and it will return False otherwise. The probability of failure for this boosted algorithm is $\frac{1}{2^k}$ since the probability of failing is equal to the probability our algorithm fails for all k random vectors (each sampled independently) where the probability of failing for a given random vector is at most $\frac{1}{2}$ □

Problem 8: Quick Select.

Solution.

a. On any iteration, the while loop breaks if the PIVOT method partitions A into two arrays (A_1, A_2) such that $\frac{n}{3} \leq \textbf{len}(A_1) \leq \frac{2n}{3}$. Because A has distinct elements, the randomly selected $q \in A[1:n]$ must lie in the middle third of the sorted version of A in order for the loop to break. In other words, there are $\frac{n}{3}$ candidates for selection that result in the loop's termination. Because each element of A may be selected by the RANDOM method with equal probability, it follows that on any iteration, the while loop breaks with probability $\frac{1}{3}$. Let X be a random variable that indicates the number of times the while loop runs. Clearly, the loop is guaranteed to run at least once for $k > 1$. The probability that the loop runs twice is the probability that it did not break on the first iteration, but does break on the second iteration. More generally, we can express the expected number of while loop iterations as

$$\textbf{Exp}[X] = \sum_{i=0}^{\infty} \frac{i}{3} \left(\frac{2}{3}\right)^{i-1}$$

$$\textbf{Exp}[X] = \frac{1}{3} \cdot \sum_{i=0}^{\infty} i \cdot \left(\frac{2}{3}\right)^{i-1}$$

$$3 \cdot \textbf{Exp}[X] = \sum_{i=0}^{\infty} i \cdot \left(\frac{2}{3}\right)^{i-1}$$

Now, consider the following property of a convergent geometric series. For any $0 < x < 1$,

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$$

Differentiating both sides with respect to x, we have

$$\sum_{i=0}^{\infty} i \cdot x^{i-1} = \frac{1}{(1-x)^2}$$

With $x = \frac{2}{3}$, this is exactly the summation that describes the expected value of our random variable X. Using this identity, it follows that

$$\begin{aligned} 3 \cdot \mathbf{Exp}[X] &= \frac{1}{(1-\frac{2}{3})^2} \\ 3 \cdot \mathbf{Exp}[X] &= 9 \\ \mathbf{Exp}[X] &= 3 \end{aligned}$$

We expect that, on any QUICKSELECT call with $k > 1$, the while loop runs 3 times. Because the PIVOT function makes no more than n comparisons, the expected number of comparisons made in the while loop can be bounded at $3n$. Note that this derivation agrees with the probability theory. Our random variable X is computing the number of independent trails before we see a successful event, where in our application the independent events are sampling from $[n]$ and a successful trial is picking an element in the array which lies middle third. This good event occurs with probability $\frac{1}{3}$. Thus, X is a geometric random variable with $p = \frac{1}{3}$ and we know the mean of these random variables is $\mathbf{E}[X] = \frac{1}{p} = 3$ □

b. $T(n)$ is defined as:

$$T(n) := \max_{A[1:n]} \mathbf{Exp}[T_Q(A)]$$

where $T_Q(A)$ is the number of comparisons on array A by Quick Select. Note that we care only about the array A that results in a maximum expected value for $T_Q(A)$. In other words, we are looking for the worst case number of comparisons across any A. First, observe that for an array of length n, the size of the subarray passed to any recursive call to QUICKSELECT can be no greater than $\frac{2n}{3}$. This is due to the fact that the initial while loop does not terminate until we select a pivot q whose index in the sorted array lies in $[\frac{n}{3}, \frac{2n}{3}]$. In the worst case, q lies at exactly $\frac{n}{3}$ and the kth smallest element is greater than q, or q lies at exactly $\frac{2n}{3}$ and the kth smallest element is less than q. Hence, we can upper bound the total expected number of comparisons by assuming that one of these two cases always occurs. Let $T(m) = T(2n/3)$. Then, $T(n) \leq T(m) + 3n$, where $T(m)$ captures the worst recursive case and $3n$ is the expected number of comparisons made within the while loop. Expanding this out, we have

$$\begin{aligned} T(n) &\leq 3n + 3\left(\frac{2n}{3}\right) + 3\left(\frac{4n}{9}\right) + 3\left(\frac{8n}{27}\right) + ... \\ &\leq \sum_{i=0}^{\infty} 3n \cdot \left(\frac{2}{3}\right)^i \\ &= 3n \cdot \sum_{i=0}^{\infty} \left(\frac{2}{3}\right)^i \end{aligned}$$

Again, we have a convergent geometric series:

$$\begin{aligned} 3n \cdot \sum_{i=0}^{\infty} \left(\frac{2}{3}\right)^i &= \frac{3n}{1-\frac{2}{3}} \\ &= 9n \end{aligned}$$

Thus, we can say that the expected worst case number of comparisons made by QUICKSELECT on an array of size $\leq n$ is at most $9n$, i.e $T(n) \leq 9n$. □

Problem 2: Handling Both Under and Over Estimates.

Solution.
Suppose we use the same strategy as in Problem 1 and take the minimum across m independent samples $est_1, est_2, ..., est_m$. In this scenario, there are two outcomes that result in us choosing a "bad" estimate: either all estimates were overestimates, or at least one estimate was an underestimate. Let X and Y denote these events, respectively. In particular, define

$$\begin{aligned} X &= \bigwedge_{i=1}^{m} est_i \geq (1+\varepsilon)N \\ Y &= \bigvee_{i=1}^{m} est_i \leq (1-\varepsilon)N \end{aligned}$$

Since each sample is chosen independently, it follows that for any est_i, $\mathbf{Pr}[est_i \geq (1+\varepsilon)N] \leq 0.9$. Therefore,

$$\begin{aligned} \mathbf{Pr}[X] &= \prod_{i=1}^{m} \mathbf{Pr}[est_i \geq (1+\varepsilon)N] \\ &\leq \prod_{i=1}^{m} 0.9 \\ &= 0.9^m \end{aligned}$$

Similarly, we have that $\mathbf{Pr}[est_i \leq (1-\varepsilon)N] \leq 0.01$. This implies that

$$\begin{aligned} \mathbf{Pr}[Y] &= \sum_{i=1}^{m} \mathbf{Pr}[est_i \leq (1-\varepsilon)N] \\ &\leq \sum_{i=1}^{m} 0.01 \\ &= 0.01m \end{aligned}$$

Let Z be the event that, using this strategy, we obtain either an underestimate or an overestimate. Because X and Y are mutually exclusive events, it follows that $\mathbf{Pr}[Z] = \mathbf{Pr}[X] + \mathbf{Pr}[Y]$. From the above, we have that $\mathbf{Pr}[Z] \leq 0.9^m + 0.01m$. Now, suppose $m = 20$. Then, we have

$$\begin{aligned}\mathbf{Pr}[Z] &\leq 0.9^{20} + 0.01(20) \\ &\approx 0.3216 \\ &< \frac{1}{3}\end{aligned}$$

Thus, we can say that taking the minimum of 20 independent samples of est will result in a "bad" estimate with probability less than $\frac{1}{3}$. Let W be a random variable that takes the value 1 if, after obtaining 20 independent samples of est and choosing the minimum, the selected estimate was either an overestimate or an underestimate. For the sake of obtaining an upper bound, suppose $\mathbf{Pr}[W = 1] = \frac{1}{3}$. Then, $\mathbf{Exp}[W] = \frac{1}{3}$. Because the expected value of W is exactly equal to the statistic we wish to estimate, it follows that W is an unbiased estimator. Therefore, we can apply the Boosting/Median-of-Means theorem in order to find an (ε, δ)-estimator. Applying the theorem, we can achieve such an estimator by taking k independent samples of W, with

$$k = \frac{C\mathbf{Var}[W]}{\mathbf{Exp}[W]^2} \cdot \frac{1}{\varepsilon^2} \cdot \ln\left(\frac{2}{\delta}\right)$$

where C is some constant. Substituting our values, we have

$$\begin{aligned}k &= \frac{C\frac{2}{9}}{\frac{1}{9}} \cdot \frac{1}{\varepsilon^2} \cdot \left(\ln\frac{2}{\delta}\right) \\ &= \frac{2C}{\varepsilon^2} \cdot \ln\left(\frac{2}{\delta}\right) \\ &= O\left(\ln\left(\frac{1}{\delta}\right)\right)\end{aligned}$$

Note that each k requires 20 samples of our actual estimator. Since this is a constant factor, however, the asymptotic size of k does not change.