Name: ____________________

Entry number: ____________________

There are 3 questions for a total of 10 points.

---

1. Let $X$ be a random variable denoting the number of people attending a conference. We know that $E[X] = 100$. Everyone shakes hand with everyone else at the conference and let $Y$ denote the total number of handshakes. We know that $E[Y] = 5000$. Answer the following questions.

   (a) (1 ½ points) What is the variance of $X$? Show calculations in the space below.

   (a) 100

   **Solution:** Since $Y = \binom{X}{2}$, we have $5000 = E[Y] = E[(1/2) \cdot X(X-1)] = (1/2) \cdot E[X^2] - (1/2) \cdot E[X]$. Since, $E[X] = 100$, we get that $E[X^2] = 10100$. Now, $Var[X] = E[X^2] - (E[X])^2 = 10100 - 10000 = 100$.

   (b) (1 ½ points) Use Chebychev's inequality to give an upper bound on the probability that less than 80 people attend the conference?

   (b) $\frac{100}{441}$

   **Solution:** Using Chebychev's inequality, we get the following:

$$\begin{aligned} \mathbf{Pr}[X < 80] &\leq \mathbf{Pr}[|X - 100| \geq 21] \\ &= \mathbf{Pr}[|X - E[X]| \geq 21] \\ &\leq \frac{Var[X]}{21^2} = \frac{100}{21^2}. \end{aligned}$$

2. (4 points) Consider executing the following algorithm on an array $A[1...n]$ containing $n$ distinct numbers$\{N_1, N_2, ..., N_n\}$ permuted randomly.

```
FindMax(A, n)
  - Max ← A[1]
  - For i = 2 to n
      - If (A[i] > Max) Max ← A[i]
  - return(Max)
```

   Let $X$ be the random variable denoting the number of times the variable $Max$ is updated within the for loop of the `FindMax` algorithm. What is $\mathbf{E}[X]$ as a function of $n$? Express your answer concisely using big-Theta notation. Show your calculations in the space below.

   2. $\Theta(\log n)$

**Solution:** Let $X_j$ be the indicator random variable that is 1 if in iteration $j$ the value of $Max$ is updated and 0 otherwise. Let $X = \sum_{j=2}^{n} X_j$. In iteration $j$, the value of $Max$ is updated iff $A[j]$ is the maximum element in the subarray $A[1...j]$. The probability of this happening is $\frac{1}{j}$. So, we have $\mathbf{E}[X_j] = \frac{1}{j}$. So, from linearity of expectation, we get that $\mathbf{E}[X] = \sum_{j=2}^{n} \mathbf{E}[X_j] = \sum_{j=2}^{n} \frac{1}{j}$. Let $H_n = 1 + 1/2 + 1/3 + ... + 1/n$. The following (outlined in class) holds.

$$\begin{aligned} \int_1^n \frac{1}{x} dx \leq \quad & H_n \quad \leq 1 + \int_2^{n+1} \frac{1}{x-1} dx \\ \Rightarrow \ln n \leq \quad & H_n \quad \leq 1 + \ln n \end{aligned}$$

This gives $\mathbf{E}[X] = H_n - 1 = \Theta(\log n)$.

3. Consider the following randomized quick-sort algorithm for sorting an array $A$ containing distinct numbers:

`Randomized-Quick-Sort(`$A$`)`
- If $(|A| = 1)$return$(A)$
- Randomly pick an index $i$ in the array $A$
- Use $A[i]$ as a pivot to partition $A$ into $A_L$ and $A_R$
  *// That is, $A_L$ denotes the array of elements that are smaller than $A[i]$, and $A_R$ denotes the*
  *//array of elements that are larger than $A[i]$. The relative ordering of elements in $A_L$ (and $A_R$)*
  *//is the same as that in $A$*
- $B_L \leftarrow$ `Randomized-Quick-Sort(`$A_L$`)`
- $B_R \leftarrow$ `Randomized-Quick-Sort(`$A_R$`)`
- return$(B_L|A[i]|B_R)$

We will try to compute the expected number of pairwise comparisons performed by the algorithm during its execution. Note that comparisons are done during the pivoting operation.

(a) (1 ½ points) For $i < j$, let $X_{ij}$ denote the indicator random variable that is 1 if a comparison between $A[i]$ and $A[j]$ is done during the execution of the algorithm and 0 otherwise. What is the value of $\mathbf{E}[X_{ij}]$ in terms of $i$ and $j$? You do not need to give reasons.

(a) $\frac{2}{j-i+1}$

(b) (1 ½ points) Let $X = \sum_{i<j} X_{ij}$. Note that $X$ denotes the total number of pairwise comparisons. Use part (a) to give $\mathbf{E}[X]$ as a function of $n$. Express your answer concisely using big-Theta notation. You do not need to show calculations.

(b) $\Theta(n \log n)$

**This is for explanation. You were not expected to write this.**
After $A[i]$ and $A[j]$ get partitioned, they cannot be compared anymore. When they get partitioned, they get compared when either $A[i]$ is chosen as pivot or $A[j]$ is chosen as pivot. So, the probability of this is $\frac{2}{j-i+1}$.