Scalable Algorithms for Data Analysis

Assignment 1

Aug-Nov 2022

1 Instructions

1. Deadline: 5pm on 21-Sep-2022.
2. Maximum mark : 25
3. Solve it by yourself and be honest.
4. For each question write answers on separate sheets.
5. Submit the answers through gradescope (see the course description page in piazza).

2 Questions

1. Two sets A and B are chosen independently of each other uniformly at random from the power set of $\{1, ..., n\}$. Find
   (a) $\Pr[A \subset B]$; (1)
   (b) the probability that A and B are disjoint. (1)
2. Two people toss a fair coin n times. Find the probability that they will score the same number of heads. (1)
3. I have a fair coin and a two headed coin. I choose one of the two coins randomly with equal probability and flip it. Given that the flip was head, what is the probability that I flipped the two-headed coin? (2)
4. A permutation on the numbers $\{1, 2, \ldots, n\} = [n]$ can be represented as a function $\pi : [n] \to [n]$, where $\pi(i)$ is the position of i in the ordering given by the permutation. A fixed point of a permutation $\pi : [n] \to [n]$ is a value for which $\pi(x) = x$. Find the expected number of fixed points of a permutations chosen uniformly at random from all permutations. (2)
5. Describe a method for using a fair coin to generate a permutation π of $\{1, \ldots, n\}$ uniformly at random. What is the expected number of coin tosses of your scheme. (2)
6. A man with n keys wants to open his door and tries the key independently and at random. Find the expected number of trials. (2)
   (a) if unsuccessful keys are not eliminated from further selections;

(b) if they are.

(Assume that only one key fits the door.)

7. Suppose $X_1, X_2, \ldots, X_n$ are pairwise independent random variables over the finite probability space $(\Omega, \Pr)$. Let $X = \sum_{i=1}^{n} X_i$. Prove that $Var[X] = \sum_{i=1}^{n} Var[X_i]$. (2)

8. **Balls and bins.** Consider the standard balls and bins process. A collection of m identical balls are thrown into n bins. Each ball is thrown independently into a bin chosen uniformly at random. (3)

   (a) What is the (precise) probability that a particular bin i contains exactly k balls at the end of the experiment.

   (b) Let X be the number of bins that contain exactly k balls. What is the expected value of X?

   (c) What is the variance of X? [***Hint:*** Write $X = \sum_i Y_i$ where Y_i is indicator random variable for bin i having exactly k balls.]

9. Let σ be a uniformly random permutation of $\{1 \ldots, n\}$. That is $\sigma(1), \sigma(2), \ldots, \sigma(n)$ is a permutation and it is chosen uniformly from one of the $n!$ permutations. We say that position i is a peak in σ if $\sigma(i)$ is the maximum number amongst $\sigma(1), \sigma(2), \ldots, \sigma(i)$. For instance if σ is the permutation $3, 4, 1, 2, 5$ then positions $1, 2, 5$ are peaks and positions 3 and 4 are not. Note that position 1 is always a peak. Let σ be a uniform random permutation of $\{1 \ldots, n\}$.

   (a) What is the probability that position i is a peak in σ? (2)

   (b) What is the expected number of peaks in σ? (2)

10. Consider the following variant of Quick Sort. Given an array A of n numbers (which we assume are distinct for simplicity) the algorithm picks a pivot x uniformly at random from A and computes the rank of x. If the rank of x is between $n/4$ and $3n/4$ (call such a pivot a good pivot) it behaves like the normal Quick Sort in partitioning the array A and recursing on both sides. If the rank of x does not satisfy the desired property (the pivot picked is not good) the algorithm simply repeats the process of picking the pivot until it finds a good one. Note that in principle the algorithm may never terminate! (5)

    (a) Write a formal description of the algorithm.

    (b) Prove that the expected runtime of this algorithm is $O(n \log n)$ on arrays of n numbers.

    (c) What is the expected number of peaks in σ?

    (d) Prove that the algorithm terminates in $O(n \log n)$ time with high probability. Does this immediately imply that the expected run time is $O(n \log n)$?