CS648 : Randomized Algorithms
Semester II, 2016-17, CSE, IIT Kanpur

Practice sheet 1

The topics are:

- Basic concepts of probability theory
- Linearity of expectation

1. **bins with i balls**

   We throw m balls randomly, uniformly, and independently into n bins. What is the expected number of empty bins ? What is the expected number of bins containing exactly i balls.

2. **Sum of samples**

   An urn contains n balls numbered 1,2,...,n. We remove k balls at random (without replacement) and add up their numbers. Find the expected value of this final number.

3. **Surviving couples**

   Of the $2n$ people in a given collection of n couples, exactly m die. Assuming that the m have been picked at random, find the expected number of surviving couples.

4. **Number of runs in a coin toss**

   A biased coin is tossed n times, and heads shows with probability p on each toss. A *run* is a sequence of throws which result in the same outcome, so that, for example, the sequence **HHTHTTH** contains five runs. What is the expected number of runs in a sequence of n tosses ?

5. **Magnet blocks**

   A total of n bar magnets are placed end to end in a line with random independent orientations. Adjacent like poles repel, ends with opposite polarities join to form blocks. Find the expected number of blocks of joined magnets.

6. **Stick break**

   Given a stick with n joints. The stick is dropped from certain height. During the fall, each joint breaks with probability p independent of other joints. What is the expected number of pieces into which the stick breaks ?

7. **Memoryless Guessing**

   To amaze your friends, you have them shuffle a deck of n cards and then turn over one card at a time. Before each card is turned over, you predict its identity. Unfortunately, you don't have any particular psychic abilities - and you are not good at remembering what has been turned over already- so your strategy is simply to guess a card uniformly at random from the deck. What is the expected number of correct predictions that you would make with this strategy ?

8. Let X and Y be two independent random variables defined over a probability space (Ω, P). Prove that

   $$\mathbf{E}[X \cdot Y] = \mathbf{E}[X] \cdot \mathbf{E}[Y]$$

   (you are advised to internalize the proof).

   the expected total distance that element will have to be moved.

9. Suppose that n balls are thrown independently and uniformly at random into n bins.

   (a) Find the conditional probability that bin 1 has one ball given that exactly one ball fell into the first three bins.

   (b) Find the conditional probability that the number of balls in bin 1 is 2 under the condition that bin 2 received $n/2$ balls.

   (c) What is the conditional probability that nth bin is empty given that the bins numbered 1 to $n/2$ are empty ?

10. Recall the randomized quick sort algorithm. Let e_i denote ith smallest element in the array. Let X_{ij} be the random variable which takes value 1 if e_i is compared with e_j, and zero otherwise.

    Prove that the random variables $\{X_{ij} | 1 \leq i < j \leq n\}$ are not independent. In more precise words, demonstrate a few random variables which are not independent.

11. Recall the "balls into bins" experiment.

    - Let X_i be the random variable which denote the number of balls choosing ith bin. Are the random variables X_i's independent ?
    - Let Y_i be the random variable which denotes the destination bin for ith ball. Are the random variables Y_i's independent ?

12. There is a bag containing r red balls and b blue balls. We take out balls one by one, uniformly randomly and throw them away. During each step, every ball present in the bag is equally likely to be removed. What is the expected number of red balls left after all the black balls have been taken out ?

13. We know various height balanced binary search trees (Red-Black trees or AVL trees). Each of these trees keep an additional balance field in each node. We also know that the procedure to keep the tree balanced is quite tedious (it takes 1 or 2 lectures to explain all cases for insertions and deletions). Here is a randomized variant of binary search tree.

    Let S be the set of elements for which we wish to build a binary search tree. Build the tree incrementally as follows: Select and remove a uniformly random element from S and make it the root. Now repeat the following step until S becomes empty: Select and remove a randomly and uniformly selected element from S and insert it into the present binary search tree.

    - Prove that the expected depth (distance from the root) of each element in the tree will be $O(\log n)$.

Note: This practice sheet should be solved over a span of 10 days. There are a few questions in this sheet which were asked during the lectures.