Problem Set I

1. Suppose G is a graph with vertiex set $V = \{1, 2, .., n\}$ and edge set $E = \{e_1, e_2, ..e_m\}$. Suppose each vertex is given colour red with probability p and blue with probability $1 - p$ independently.

    1. What is the expected number of vertices coloured red.
    2. What is the probability that a given edge has both end points of the same colour?
    3. What is the expected number of edges whose one end point is red and the other end point blue? (Note that this defines a cut in the graph.)
    4. What is the value of p that maximizes the expected cut size? (Use calculus to maximize the expectation).

2. Suppose there are k bins. We throw balls uniformly at random into these bins so that each ball is likely to land into any of the bins with equal probability.

    1. What is the expected number of throws for the first bin gets a ball?
    2. What is the expected number of throws needed for the first and the second bin to be occupied?
    3. What is the expected number of throws need for two bins to get occupied. (Here, we are waiting for the event that that after the first throw, a ball lands in a bin different from the bin to which the first throw landed).
    4. What is the expected number of throws needed for three bins to get occupied?
    5. What is the expected number of throws needed to reach a condition that every bin contains at least one ball?

3. An array of n distinct elements is given.

    1. Suppose you want to search for an element in the array, but you don't know its position. Suppose you try repeatedly checking a random position (independently) and testing whether the number in that position is the one you are searching for, what is the expected number of steps needed to hit the correct position?
    2. Suppose instead, you simply pick a random element in the array and print it out iteratively. How may iterations are required (on the average) to print every element in the array at least once?
    3. Do you see any connection between this question and the previous question?

4. A particle does a *random walk* from position $n > 0$ in the following wa at each step, If the current position is n, it moves in the next step to position $n-1$. If the current position in $n > i > 0$, in the next step it moves to position either $i + 1$ or $i - 1$ with equal probability. What is the expected number of steps necessary for the particle to reach position 0? (Let X_i denote the random variable that measures the number of steps required for the particle to walk from i to reach 0. Clearly $X_0 = 0$. $X_n = 1 + X_{n-1}$, defining the boundary conditions. Set up a set of probabilistic recurrance equations for modeling the problem. Set $T(i) = E(X_i)$ and arrive the equation $T(i) = 1 + \frac{1}{2}[T(i-1) + T(i+1)]$ for $0 < i < n$. Solve the recurrances and prove that $T(n) = O(n^2)$.

5. A 2-CNF formula is a boolean formula in conjunctive normal form with exactly two literals per clause, with the condition that variables in a clause are different. For example, $(x_1 \vee x_2) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_2 \vee x_3)$ is a 2-CNF formula in 3 variables x_1, x_2, x_3 containing 3 clauses. Note that each clause is a disjunction of literals (a literal is a variable or a negated variable - like x_1, $\neg x_1$ etc.) and the formula is a conjunction of clauses.

    Suppose you are given a 2-CNF formula in n variables and k clauses. A satisfying truth assignment to the formula is a truth assignment to the variables that satisfies each clause in the formula. Suppose you are given a 2-CNF formula such that it has at least one satisfying truth assignment. In this question, we will develop an algorithm to find such a satisfying assignment starting from an arbitrary truth assignment.

    We say two truth assignments α and β are at (Hamming) distance k if these two truth assignments differ only in the truth values of exactly k variables. Suppose now, we change the truth value of one of the variables in α, then the resultant truth assignment will be either at a distance $k+1$ or $k-1$. Thus changing one variables either increases or decreases the distance by one. The single exeception to this rule is when one of the truth assignments is that the "all False" assignment and the other is the "all True assignment". In this case, α and β are already at maximum distance and any value change can only bring them closer by one step.

Here is the algorithm. Start with the all zero truth assignment. If all clauses are satisfied, we are done. Otherwise, pick any clause that is not satisfied by this assignment. Clearly, we need to flip one of these variables, to reach the correct truth assignment; but we don't know which one. So, we randomly flip the value of one of the variables. Thus, we may assume that unless we are maximally far apart already from the goal, with probability at least half, we more closer one step. We keep repeating this experiment till a satisfying truth assignment is obtained.

Find the expected number of flips needed to reach a satifying truth assignment. Assume worst case distance initially. (Observe the connection with the previous question).