E0 225 Design & Analysis of Algorithms **Assignment - 7**
Sai Phanindra Korlepara — **24142** — M.Tech CSA Dept.

Problem 1. *KT Book, Chapter 13, Problem 2*
Consider a county in which 100,000 people vote in an election. There are only two candidates on the ballot: a Democratic candidate (denoted D) and a Republican candidate (denoted R). As it happens, this county is heavily Democratic, so 80,000 people go to the polls with the intention of voting for D, and 20,000 go to the polls with the intention of voting for R.

However, the layout of the ballot is a little confusing, so each voter, independently and with probability 1/100, votes for the wrong candidate— that is, the one that he or she didn't intend to vote for. (Remember that in this election, there are only two candidates on the ballot.)

Let X denote the random variable equal to the number of votes received by the Democratic candidate D, when the voting is conducted with this process of error. Determine the expected value of X, and give an explanation of your derivation of this value.

Solution.

We are given that the random variable X denotes the number of votes received by the democratic candidate. The total Expected number of votes that the democratic candidate will be getting can be characterized as follows.

Let p_d denote the probability for a voter being democratic and p_r denote the probability that a voter is republican. Similarly let p_c denote the probability that a voter votes to their intended recipient, p_f denote the probability that a voter votes to candidate that is not their intended recipient.

$$X = Population \times (p_d \times p_c + p_r \times p_f)$$

$$X = 100000(0.8 \times 0.99 + 0.2 \times 0.01 = 79400$$

Hence the expected number of votes received by the democratic candidate is 79,400. □

Problem 2. *KT Book, Chapter 13, Problem 3*
Question redacted for being too long.

Solution.

1. Each process picks 1 with probability 1/2 and 0 with probability 1/2. The process succeeds if all the processes in it's conflict list choose 0.

   In this scenario:

   Each process P_i has a random choice for x_i where:

   $x_i = 1$ with probability $\frac{1}{2}$,

   $x_i = 0$ with probability $\frac{1}{2}$.

   P_i will enter the set S if and only if it chooses $x_i = 1$ and all conflicting processes (those it "competes" with) choose $x = 0$.

   Given, n the total number of processes, d the number of conflicts per process.

Analysis of Probability for Process P_i Entering Set S

The probability that each conflicting process chooses $x = 0$ is $1/2$. Thus, for all d conflicting processes to choose $x = 0$, the probability is $(1/2)^d$.

Therefore, the probability that P_i will enter S is:

$$\Pr(P_i \in S) = \frac{1}{2} \times \left(\frac{1}{2}\right)^d = \frac{1}{2^{d+1}}$$

The expected number of processes that enter the set S is:

$$E[\text{Processes in } S] = n \times \Pr(P_i \in S) = n \times \frac{1}{2^{d+1}} = \frac{n}{2^{d+1}}$$

This gives the expected count of processes that will enter the set S.

2. In this section we need to replace the probability that a process chooses $x_i = 1$ to be p, $x_i = 0$ to be 1-p. we can rewrite the probability and expectation as follows,

$$\Pr(P_i \in S) = p \times (1-p)^d = p(1-p)^d$$

The expected number of processes that enter the set S is:

$$E[\text{Processes in } S] = n \times \Pr(P_i \in S) = n \times p(1-p)^d = np(1-p)^d$$

We need to find a value for p that maximizes the above expectation. To do this we can find the first derivative and equate to zero to find the stationary points. Let $f(p) = np(1-p)^d$

$$f'(p) = 0 = n(1-p)^d - ndp(1-p)^{d-1}$$
$$(1-p)^d = dp(1-p)^{d-1}$$
$$1 - p = dp$$
$$p = 1/(d+1)$$

This gives the value for p which is the only stationary point. We can verify the second derivative to see if it's a maxima or a minima.

$$f''(p) = -2nd(1-p)^{d-1} + nd(d-1)p(1-p)^{d-2} = -nd(1-p)^{d-2}\left[2(1-p) - p(d-1)\right]$$

To continue, we evaluate $f''(p)$ at $p = \frac{1}{d+1}$. With $1 - p = \frac{d}{d+1}$, we substitute into $f''(p)$:

$$f''\left(\frac{1}{d+1}\right) = -nd\frac{d^{d-1}}{(d+1)^{d+1}}$$

Since $f'' < 0$, this confirms that $p = 1/(d+1)$ maximizes $f(p)$. Thus, the optimal p for maximizing the expected number of processes in S is $p = 1/(d+1)$.

□

Problem 3. *KT Book, Chapter 13, Problem 6*
One of the (many) hard problems that arises in genome mapping can be formulated in the following abstract way. We are given a set of n markers $\mu_1, ..., \mu_n$—these are positions on a chromosome that we are trying to map—and our goal is to output a linear ordering of these markers. The output should be consistent with a set of k constraints, each specified by a triple (μ_i, μ_j, μ_k), requiring that μ_j lie between μ_i and μ_k in the total ordering that we produce. (Note that this constraint does not specify which of μ_i or μ_k should come first in the ordering, only that μj should come between them.)

Now it is not always possible to satisfy all constraints simultaneously, so we wish to produce an ordering that satisfies as many as possible. Unfortunately, deciding whether there is an ordering that satisfies at least k' of the k constraints is an NP-complete problem (you don't have to prove this.)

Give a constant α¿0 (independent of n) and an algorithm with the following property. If it is possible to satisfy k^* of the constraints, then the algorithm produces an ordering of markers satisfying at least αk^* of the constraints. Your algorithm may be randomized; in this case it should produce an ordering for which the expected number of satisfied constraints is at least αk^*.

Solution. This is a problem of maximum satisfiability. Since there is no requirement that is placed on the constant α, We can design a random algorithm that satisifes the following property.

Algorithm

Generate a random ordering of the markers that are provided.

For any given constraint (μ_i, μ_j, μ_k), In a random sequence where μs can be strewn away randomly. There are 6 possible orderings (3!) of these 3 markers. Only 2 of these orderings (The two that have μ_j in the middle) satisfy the given constraint. This gives the probability that any given constraint is satisfied in a random sequence of the markers as 1/3.

If there are a total of k^* maximum possible constraints that can be satisfied, the expected number of constraints that a random sequence of the markers is $k^*/3$. Hence if we set $\alpha = 1/3$, we can say that the expected number of constraints that are satisfied by this randomized algorithm is αk^*. □

Problem 4. *KT Book, Chapter 13, Problem 7a*
First consider the randomized approximation algorithm we used for MAX3-SAT, setting each variable independently to true or false with probability 1/2 each. Show that the expected number of clauses satisfied by this random assignment is at least k/2, that is, at least half of the clauses are satisfied in expectation. Give an example to show that there are MAX SAT instances such that no assignment satisfies more than half of the clauses.

Solution. The problem and algorithm given are,

Problem : We are given a set of k clauses that can have 1 to n variables per clause. A clause is satisfied if at least one of the literals is satisfied.

Algorithm : Set each variable to true with probability 1/2, set it to false with probability 1/2.

Let's say a chosen clause has i variables in it, The probability that this clause is satisfied by any given random truth assignment to the variables is, $1 - (1/2)^i$. We are subtracting from 1, the probability that all the assignments failed.

The value of i can vary from 1 to n, this makes the probability of clause being satisfied maximum when i = n where p = $1 - (1/2)^n$. The probability is lowest when there is only 1 variable in clause, here probability becomes p = 1/2.

If there are total k clauses in the problem, the worst expected number of clauses to be satisfied = $k \times \textit{worst probability} = k/2$. Hence we have shown that even for the worst possible case the expected number of clauses to be satisfied is $k/2$.

As an example let's consider a instance of MAX-SAT problem with 1 variable say x, and the following 2 clauses on it

1. $C_1 = x$
2. $C_2 = \neg x$

In this instance of the MAX-SAT problem no matter the truth value chosen for the variable x, we can only satisfy 1 of the 2 clauses, hence this problem can never satisfy, no matter the truth assignment, more than 1/2 of the clauses. □
