(CS1.406) Advanced Algorithms (Spring 2024) Due: 05.03.2024

Problem Set 2

Instructions:

- Discussions amongst the students are not discouraged, but all writeups must be done individually and must include names of all collaborators.
- Referring sources other than the lecture notes is discouraged as solutions to some of the problems can be found easily via a web search. But if you do use an outside source (eg., text books, other lecture notes, any material available online), do mention the same in your writeup. This will not affect your grades. However dishonesty of any sort when caught shall be heavily penalized.
- Be clear in your arguments. Vague arguments shall not be given full credit.

---

1. Let $a_1, a_2, \ldots, a_n$ be a list of n distinct numbers. We say that a_i and a_j are inverted if $i < j$ but $a_i > a_j$. The bubblesort algorithm swaps pairwise adjacent inverted numbers in a list until there are no more inversions, so the list is in sorted order. Suppose that the input to Bubblesort is a random permutation, equally likely to be any of the $n!$ permutations of n distinct numbers. Determine the expected number of inversions that need to be corrected by Bubblesort.

2. Let G be a graph on n vertices, with $\frac{nd}{2}$ edges. Consider the following probabilistic experiment for finding an independent set in G. Delete each vertex of G (together with its incident edges) independently with a probability of $1 - \frac{1}{d}$.

   (a) Compute the expected number of vertices and edges that remain after the deletion process.

   (b) From these, infer that there is an independent set with at least $\frac{n}{2d}$ vertices in any graph on n vertices with $\frac{nd}{2}$ edges.

3. Let $g(x) \equiv x \mod n$. For each $a \in \mathbb{Z}_p$, define the function $f_a(x) \equiv a \cdot x \mod p$, and $h_a(x) = g(f_a(x))$, and let $H = \{h_a \mid a \in \mathbb{Z}_p, a \neq 0\}$. Show that H is such that for all $x \neq y$,

$$\mathbb{P}\left[\text{There exists } h \in H \text{ such that } h(x) = h(y)\right] \leqslant \frac{2\,|H|}{n}.$$

4. Recall the bit-fixing scheme for *Oblivious Permutation Routing* from the class.

   Let G be a hypercube on n bits. For vertices $u, v \in \{0,1\}^n$, bit-fixing scheme decides a path from $u \rightsquigarrow v$ by scanning the bit strings corresponding to u and v from the most significant bit to the least significant bit and flips the individual bits in order. That is, correct the mismatched bits in order.

   Now consider the following variant of the bit-fixing scheme. Each packet p_v (for $v \in \{0,1\}^n$) randomly orders the position of bits in the string of its source v and corrects the mismatched bits in order (of this new random string). Show that there is a permutation for which with high probability this algorithm requires $2^{\Omega(n)}$ steps to complete the routing.

5. Consider the problem of deciding whether two integer multisets S_1 and S_2 are identical in the sense that each integer occurs the same number of times in both sets. The problem can be solved by sorting the two sets in $O(n \log n)$ time where n is the cardinality of the multisets. Suggest a way of representing this as a problem involving verification of polynomial identity and thereby obtain an efficient randomized algorithm.

6. Let m, n be large integers such that $m > n$. For an integer q, $\mathbb{Z}_q$ is the set of integers modulo q with the operations of addition and multiplication.

   Let the problem Max2Lin($\mathbb{Z}_q$) be defined as follows: given a system S of m linear equations over n variables, each of which depends on at most two variables, whatis the maximum number of linear equations that can be satisfied over all the assignments to the variables from $\mathbb{Z}_q$.

   Formally, let $X = \{x_1, \ldots, x_n\}$ and $S = \{L_1(X) = 0, \ldots, L_m(X) = 0\}$. For all $i \in [m]$, each $L_i(X) = 0$ is of the form

   $$c_{i_1} x_{i_1} + c_{i_2} x_{i_2} + c_{i_3} \equiv 0 \mod q \quad \text{for some constants } c_{i_1}, c_{i_2} \text{ and } c_{i_3}.$$

   (a) Provide a randomized scheme for the assignment of the variables such that the expected number of linear equations that can be satisfied is $\frac{m}{q}$.

   (b) Show that we can deterministically obtain an assignment that satisfies at least $\frac{m}{q}$ linear equations in S.

Extra problems

1. Let n be an asymptotically large integer. Using the probabilistic method show that there is a bipartite graph $G = (L, R, E)$ with the following properties.

   - $|L| = |R| = n$.
   - Each vertex in L has a degree of $n^{3/4}$ and each vertex in R has a degree of at most $3n^{3/4}$.
   - Every subset of $n^{3/4}$ vertices in L has at least $n - n^{3/4}$ neighbours in R.

1. Let $a_1, a_2, \ldots, a_n$ be a list of n distinct numbers. We say that a_i and a_j are inverted if $i < j$ but $a_i > a_j$. The bubblesort algorithm swaps pairwise adjacent inverted numbers in a list until there are no more inversions, so the list is in sorted order. Suppose that the input to Bubblesort is a random permutation, equally likely to be any of the $n!$ permutations of n distinct numbers. Determine the expected number of inversions that need to be corrected by Bubblesort.

Let $X_{ij} = \begin{cases} 1 & \text{if } i<j \text{ but } a_i > a_j \\ 0 & \text{otherwise} \end{cases}$

Let I = total no. of inversions $= E\left[\sum_{i<j} X_{ij}\right]$

We need to find $E(I)$ for a random permutation.

For a given pair i, j and a random permutation, either $a_i < a_j$ in which case there is no inversion. If $a_i > a_j$, there is an inversion.

Both of these outcomes are equally likely.

$\therefore \Pr(X_{ij} = 1) = \frac{1}{2} = \Pr(X_{ij} = 0)$

$\Rightarrow E(X_{ij}) = 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = \frac{1}{2}$

$\therefore E\left(\sum_{i<j} X_{ij}\right) = \sum_{i<j} E(X_{ij}) = \sum_{i<j} \frac{1}{2}$ ← Linearity of Expectation

Now, the no. of pairs s.t. $1 \le i < j \le n$

$= n-1 + n-2 + n-3 + \ldots + 1 + 0 = \frac{n(n-1)}{2}$

$\therefore E(I) = \frac{1}{2} \cdot \sum_{i<j} 1 = \frac{n(n-1)}{4}$

2. Let G be a graph on n vertices, with $\frac{nd}{2}$ edges. Consider the following probabilistic experiment for finding an independent set in G. Delete each vertex of G (together with its incident edges) independently with a probability of $1 - \frac{1}{d}$.

   (a) Compute the expected number of vertices and edges that remain after the deletion process.

   (b) From these, infer that there is an independent set with at least $\frac{n}{2d}$ vertices in any graph on n vertices with $\frac{nd}{2}$ edges.

(a) Let V_i be the random variable s.t.

$$V_i = \begin{cases} 1 & \text{if } i^{th} \text{ node survives after the deletion process} \\ 0 & \text{otherwise} \end{cases}$$

Probability that a node survives (i.e. is not deleted) $= 1 - (1 - \frac{1}{d})$

$= \frac{1}{d}$.

$\therefore Pr(V_i = 1) = \frac{1}{d} \quad \forall\, i \in [n]$

$\therefore E(V_i) = \frac{1}{d}$.

Let V_T = Total no. of nodes survived

$\therefore V_T = \sum_{i=1}^{n} V_i$

→ Linearity of Expectation

$\therefore E(V_T) = E(\sum_{i=1}^{n} V_i) = \sum_{i=1}^{n} E(V_i) = \sum_{i=1}^{n} \frac{1}{d} = \frac{n}{d}$.

Let $m = nd/2$ be the no. of edges in the graph.

Let e_i be the random variable such that:

$$e_i = \begin{cases} 1 & \text{if the } i^{th} \text{ edge survives after the deletion process.} \\ 0 & \text{otherwise} \end{cases}$$

A given edge survives if and only if both of the nodes making up that edge survive.

$$\therefore \Pr(e_i = 1) = \frac{1}{d} \times \frac{1}{d} = \frac{1}{d^2}$$

(Prob. of node surviving)

$$\therefore E(e_i) = 1 \cdot \Pr(e_i = 1) + 0 \cdot \Pr(e_i = 0)$$

$$= 1 \cdot \frac{1}{d^2} + 0 \cdot \Pr(e_i = 0) = \frac{1}{d^2}.$$

Let e_T be the total no. of edges surviving after the the deletion process.

Clearly, $e_T = \sum_{i=1}^{m} e_i$

$$\therefore E(e_T) = E\left(\sum_{i=1}^{m} e_i\right) = \sum_{i=1}^{m} E(e_i) = \sum_{i=1}^{m} \frac{1}{d^2} = \frac{m}{d^2}$$

(Linearity of Expectation)

Here, $m = nd/2$

$$\therefore E(e_T) = \frac{\frac{nd}{2}}{d^2} = \frac{n}{2d}.$$

(b) Consider the algorithm below:

1) Delete each node from G with probability $1 - \frac{1}{d}$. Let the after the deletion process be G'.

2) For each edge in Graph G', pick one of the nodes making up the edge uniformly at random and delete it.

Since for each edge, at most 1 node can be deleted

The no. of nodes that survive after the second deletion process V_I (say) is $\geq V_T - e_T$

$\therefore \quad V_I \geq V_T - e_T$

$\therefore \quad E(V_I) \geq E(V_T - e_T) = E(V_T) - E(e_T)$

$= \frac{n}{d} - \frac{n}{2d} = \frac{n}{2d}.$

$\therefore \quad E(V_I) \geq \frac{n}{2d}$

The nodes remaining after the second deletion process form an Independent set of size V_I.

since $E(V_I) \geq \frac{n}{2d}$

$\therefore \quad Pr\left(V_I \geq \frac{n}{2d}\right) > 0$

} Lemma 6.2, Mitzenmacher Page. 138.

Lemma 6.2: *Suppose we have a probability space $\mathcal{S}$ and a random variable X defined on $\mathcal{S}$ such that $\mathbf{E}[X] = \mu$. Then $\Pr(X \geq \mu) > 0$ and $\Pr(X \leq \mu) > 0$.*

Proof: We have

$$\mu = \mathbf{E}[X] = \sum_x x \Pr(X = x),$$

where the summation ranges over all values in the range of X. If $\Pr(X \geq \mu) = 0$, then

$$\mu = \sum_x x \Pr(X = x) = \sum_{x<\mu} x \Pr(X = x) < \sum_{x<\mu} \mu \Pr(X = x) = \mu,$$

giving a contradiction. Similarly, if $\Pr(X \leq \mu) = 0$ then

$$\mu = \sum_x x \Pr(X = x) = \sum_{x>\mu} x \Pr(X = x) > \sum_{x>\mu} \mu \Pr(X = x) = \mu,$$

again yielding a contradiction. ■

$\therefore$ There is always an independent set with at least $\frac{n}{2d}$ vertices in any Graph on n vertices with $\frac{nd}{2}$ edges.

3. Let $g(x) \equiv x \mod n$. For each $a \in \mathbb{Z}_p$, define the function $f_a(x) \equiv a \cdot x \mod p$, and $h_a(x) = g(f_a(x))$, and let $H = \{h_a \mid a \in \mathbb{Z}_p, a \neq 0\}$. Show that H is such that for all $x \neq y$,

$$\mathbb{P}\left[\text{There exists } h \in H \text{ such that } h(x) = h(y)\right] \leqslant \frac{2|H|}{n}.$$

Let $x \neq y$

we need to find the probability of $h_a(x) = h_a(y)$ $\forall\ a \in Z_p \setminus \{0\}$

$\therefore\ h_a(x) = h_a(y)$

$\Rightarrow (f_a(x) \bmod n) = (f_a(y) \bmod n)$

$\Rightarrow \left[(ax \bmod p) \bmod n\right] = \left[(ay \bmod p) \bmod n\right]$

Case I: $ax \bmod p > ay \bmod p$

$\Rightarrow a(x-y) \bmod p \equiv n\lambda \bmod p$

here since a is non-zero, $a \in [p-1]$

λ can effectively take values between 0 & $\left\lfloor \frac{p-1}{n} \right\rfloor$

$\therefore$ for there to be a collision, a and λ must be equal and $x-y=n$.

$\therefore$ This happens with Probability $\leq \frac{\frac{p-1}{n}}{p-1} = \frac{1}{n}$

Case II: $ax \bmod p < ay \bmod p$

$\therefore a(y-x) \equiv n\mu \bmod p$

Here as well, $\mu \in [0, \frac{p-1}{n}]$, $a \in [1, p-1]$ and $y-x = n$ for there to be a collision. $\therefore$ Probability of this happening $= 1/n$

$\therefore Pr(h_a(x) = h_a(y) \mid x \neq y) = \frac{1}{n} + \frac{1}{n} = \frac{2}{n}$

since there are $p-1$ such different functions

$$\Pr(\exists\, h \in H \text{ s.t. } h(x)=h(y)) \leq \frac{2(p-1)}{n} = \frac{2|H|}{n} \quad \text{using union bound}$$

4. Recall the bit-fixing scheme for *Oblivious Permutation Routing* from the class.

   Let G be a hypercube on n bits. For vertices $u, v \in \{0,1\}^n$, bit-fixing scheme decides a path from $u \rightsquigarrow v$ by scanning the bit strings corresponding to u and v from the most significant bit to the least significant bit and flips the individual bits in order. That is, correct the mismatched bits in order.

   Now consider the following variant of the bit-fixing scheme. Each packet p_v (for $v \in \{0,1\}^n$) randomly orders the position of bits in the string of its source v and corrects the mismatched bits in order (of this new random string). Show that there is a permutation for which with high probability this algorithm requires $2^{\Omega(n)}$ steps to complete the routing.

consider the permutation

$(x_1, x_2, \ldots, x_{n/2}, 0, 0, 0 \ldots 0) \longrightarrow (0, 0, 0, \ldots, x_1, x_2, \ldots, x_{n/2})$

If we do corrections of the in random order to get $(0,0,0,\ldots,x_1,x_2,\ldots,x_{n/2})$, we would atleast require 2 times the no. of 1s in $x_1, x_2, \ldots, x_{n/2} = x$

Let the no. of 1s in $x_1, x_2, x_3, \ldots, x_{n/2}$ be ℓ.

Then minimum no. of corrections needed to reach the destination will be 2ℓ.

while the first ℓ bits are in order set, the packet will pass through $(0,0,0\ldots,0)$ (n-tuples)

$\therefore$ E(no. of packets with first $n/2$ set, and rest 0, going through origin)

$= \sum \Pr[\text{Picking first } \ell \text{ bits in } x \text{ first}]$

$= \sum \frac{1}{\binom{2\ell}{\ell}}$

The summation here is over all packets where the first $n/2$ bits are set and ℓ of them are 1.

The no. of such packets are $\binom{n/2}{\ell}$

* inequality discussed with Shiven Sinha

$\therefore$ E(Packets crossing origin) = $\dfrac{\binom{n/2}{\ell}}{\binom{2\ell}{\ell}}$

for combination,

$$ {}^{n}C_{K} = \frac{n}{K} \cdot \frac{n-1}{K-1} \cdot \frac{n-2}{K-2} \overset{K\text{-times}}{\cdots} \frac{n-K+1}{1} \geq \frac{n}{K} \cdot \frac{n}{K} \overset{K\text{-times}}{\cdots} \frac{n}{K} = \left(\frac{n}{K}\right)^{K} $$

similarly, we can show

$$ {}^{n}C_{K} < \text{Len}_{K} = \left(\frac{\text{Len}}{K}\right)^{K} < \left(\frac{en}{K}\right)^{K} \; * $$

$$ \therefore \left(\frac{n}{K}\right)^{K} \leq \binom{n}{K} \leq \left(\frac{en}{K}\right)^{K} $$

$$ \therefore \frac{\binom{\frac{n}{2}}{\ell}}{\binom{2\ell}{\ell}} \geq \left(\frac{n}{2\ell}\right)^{\ell} \cdot \left(\frac{\ell}{2e\ell}\right)^{K} = \left(\frac{n}{4e\ell}\right)^{\ell} $$

we set $\ell = \frac{n}{8e}$ to get

$$ \left(\frac{n}{4e\ell}\right)^{\ell} = \left(\frac{n}{4e\,\frac{n}{8e}}\right)^{\frac{n}{8e}} = (2)^{n/8e} = 2^{-\Omega(n)} $$

$\therefore$ the permutation would have atleast $2^{-\Omega(n)}$ steps.

5. Consider the problem of deciding whether two integer multisets S_1 and S_2 are identical in the sense that each integer occurs the same number of times in both sets. The problem can be solved by sorting the two sets in $O(n \log n)$ time where n is the cardinality of the multisets. Suggest a way of representing this as a problem involving verification of polynomial identity and thereby obtain an efficient randomized algorithm.

We can represent an integer multiset as polynomial by representing an arbitrary integer of the multiset a and its frequency f as a term in the polynomial whose power is a and whose coefficient is f i.e. $f \cdot x^a$.

∴ The algorithm for constructing polynomial from multisets is as follows:

CONSTRUCT-POLYNOMIAL(S): (S → multiset)

1. Initialize $P \leftarrow \Phi$
2. For each $a \in S$:
2.1. P.add(S.a.value, S.a.frequency)
3. Return P

Here we consider:

S → multiset containing objects as element

S.a → object representing an arbitrary element of the multiset

S.a.value → value of the integer that the object represents in the integer multiset

S.a.frequency → frequency of the integer represented by a.

P → Polynomial, initialized 0 at the beginning

P.add(m,n) → A method to add a term with power m and frequency n to P

Now that we can convert our two multisets S_1 and S_2 (say) to P_1 and P_2 (say), our problem has been reduced to verifying whether the two polynomials P_1 and P_2 are equal or not.

This can be done via Schwartz-Zippel lemma* which is as follows:

Let $Q(x_1, x_2, \ldots, x_n)$ be a non-zero multivariate polynomial of total degree d defined over the field $F[x_1, x_2, \ldots, x_n]$.

Fix any set $S \subseteq F$ and let $r_1, r_2 \ldots, r_n$ be chosen independently and uniformly at random from S. Then

$$Pr[\, Q(r_1, r_2, \ldots, r_n) = 0 \mid Q(x_1, x_2, \ldots x_n) \not\equiv 0 \,] \leq \frac{d}{|S|}$$

*Reference: Motwani, Raghvan, Prabhakar - Randomized Algorithms Page: 165

The lemma/theorem can be proved via induction.

∴ our final algorithm for verification is as follows:

VERIFY-EQUAL(S_1, S_2):

1. $P_1 \leftarrow$ CONSTRUCT-POLYNOMIAL(S_1)
2. $P_2 \leftarrow$ CONSTRUCT-POLYNOMIAL(S_2)
3. Sample $r_1, r_2, \ldots, r_n$ independently and uniformly at random from $S \subseteq F$, with $|S|$ being sufficiently large
4. Compute $P_1(r_1, r_2, \ldots, r_n)$ and $P_2(r_1, \ldots, r_n)$

5. If $P_1(r_1, r_2 \ldots, r_n) \neq P_2(r_1, r_2, \ldots r_n)$:
RETURN $P_1 \not\equiv P_2$

else:
RETURN $P_1 \equiv P_2$ with probability $> 1 - \frac{d}{|S|}$.

We can reduce our error probability by taking an arbitrarily large S or by repeating the algorithm K more times which would make our error probability $\left(\frac{d}{|S|}\right)^K$ since trials are independent from each other.

Complexity:

- The polynomial construction takes $O(n)$ time since each element in the multiset contributes one term to the polynomial.
- The evaluation of the polynomial at a point can be efficiently done in $O(n)$.
- Therefore, the overall complexity will be $O(n)$, which more efficient than the $O(n \log n)$ time required for sorting based approach.

6. Let m, n be large integers such that $m > n$. For an integer q, $\mathbb{Z}_q$ is the set of integers modulo q with the operations of addition and multiplication.

   Let the problem Max2Lin($\mathbb{Z}_q$) be defined as follows: given a system S of m linear equations over n variables, each of which depends on at most two variables, whatis the maximum number of linear equations that can be satisfied over all the assignments to the variables from $\mathbb{Z}_q$.

   Formally, let $X = \{x_1, \ldots, x_n\}$ and $S = \{L_1(X) = 0, \ldots, L_m(X) = 0\}$. For all $i \in [m]$, each $L_i(X) = 0$ is of the form

$$c_{i_1} x_{i_1} + c_{i_2} x_{i_2} + c_{i_3} \equiv 0 \mod q \quad \text{for some constants } c_{i_1}, c_{i_2} \text{ and } c_{i_3}.$$

   (a) Provide a randomized scheme for the assignment of the variables such that the expected number of linear equations that can be satisfied is $\frac{m}{q}$.

   (b) Show that we can deterministically obtain an assignment that satisfies at least $\frac{m}{q}$ linear equations in S.

Given: A system of m linear equations

$$c_{11} x_{11} + c_{12} x_{12} + c_{13} \equiv 0 \mod q \rightarrow L_1(x)$$
$$c_{21} x_{21} + c_{22} x_{22} + c_{23} \equiv 0 \mod q \rightarrow L_2(x)$$
$$c_{31} x_{31} + c_{32} x_{32} + c_{33} \equiv 0 \mod q \rightarrow L_3(x)$$
$$\vdots \qquad\qquad \vdots$$
$$c_{m1} x_{m1} + c_{m2} x_{m2} + c_{m3} \equiv 0 \mod q \rightarrow L_m(x)$$

where c_{ij}'s are all some constants, x_{ij}'s can be any variable from the set $\{x_1, x_2, \ldots, x_n\} = X$ and q is prime number.

(a)

Randomized Scheme

Assign value to x_i from the set $\mathbb{Z}_q$ uniformly at random $\forall\ i \in [n]$.

Let S_i be the random variable s.t.

$$S_i = \begin{cases} 1 & \text{if } L_i(x) \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases}$$

Now, $L_i(x) = c_{i1} x_{i1} + c_{i2} x_{i2} + c_{i3} \equiv 0 \bmod q$

There are a total of q^2 assignments of (x_{i1}, x_{i2})

Let $x_{i2} = \lambda$, for some $\lambda \in Z_q$, then

$$c_{i1} x_{i1} + c_{i2} \lambda + c_{i3} \equiv 0 \bmod q$$

$$\Rightarrow c_{i1} x_{i1} \equiv -(c_{i2}\lambda + c_{i3}) \bmod q$$

$$\Rightarrow x_{i1} \equiv -c_{i1}^{-1}(c_{i2}\lambda + c_{i3}) \bmod q \quad *$$

$$\Rightarrow x_{i1} \equiv u \bmod q, \text{ where } u \in Z_q$$

* This step is possible since there is always a multiplicative inverse for elements in the prime integer field Z_q.

$\therefore \forall \lambda \in Z_q$, the assignment of $x_{i2} = \lambda$ will have solution for the linear equation $L_i(x)$ with assignment of $x_{i1} = u$. The opposite is also true, when $x_{i1} = u$, $x_{i2} = \lambda$ will be a satisfying solution.

$\therefore$ out of q^2 possible assignments of pair of variables (x_{i1}, x_{i2}) q are satisfying assignments.

$\therefore$ The probability that $L_i(x)$ is satisfied given a random assignment of x_{i_1}, x_{i_2} from $Z_q = \frac{q}{q^2} = \frac{1}{q}$.

$\therefore \quad Pr(S_i = 1) = \frac{1}{q}$.

$\therefore \quad E(S_i) = 1 \cdot Pr(S_i = 1) + 0 \cdot Pr(S_i = 0)$

$= \frac{1}{q}$.

Let $S = \sum_{i=1}^{m} S_i$ be total number of linear equations satisfied.

Then $E(S) = E\left(\sum_{i=1}^{m} S_i\right) \overset{\text{linearity of Expectation}}{=} \sum_{i=1}^{m} E(S_i) = \sum_{i=1}^{m} \frac{1}{q} = \frac{m}{q}$.

(b) By derandomization using method of conditional expectations, we arrive at the following deterministic algorithm to find an assignment that satisfies atleast $\frac{m}{q}$ linear equations.

Algorithm

1. Initialization: Start with no variables assigned.
2. Iterative Assignment:
   - For each variable x_i, calculate* the conditional expection of the number of satisfied for each possible value in Z_q, assuming the assignments made so far and considering each choice for x_i

- Choose the value of x_i^* s.t. it maximizes (or does not decrease) the expected no. of equation satisfied, fixing x_i to this value.

- Repeat this process for each variable until all variable are assigned.

3. Termination: Once all variables are assigned, the algorithm would guarantee the no. of satisfied equations is atleast m/q based on the method of conditional expectation.

* The calculation of conditional expectation is easily done as follows:

Let K_i be the no. of equations x_i participates in, and let $x_i = \lambda$, $\lambda \in \mathbb{Z}_q$ be the assignment of x_i, then:

$$E(S \mid x_1, x_2, \dots x_{i-1}) \leq \max_{1 \leq \lambda \leq q} \{ E(S \mid x_1, x_2, \dots, x_{i-1}, x_i = \lambda) \}$$

$$\leq K_i + E(S \mid x_1, x_2, \dots, x_i)$$

The last inequality follows by considering an assignment of x_i s.t. it satisfies all the equations it participates in.
For the equations in which x_i does not participates or partially participates, the expectation of that equation being satisfied remains $\frac{1}{q}$.

∴ The algorithm boils down to a greedy algorithm, where at each step, we try to maximize the no. of equations being satisfied. This greedy algorithm will surely give us an assignment that satisfies atleast $\frac{m}{q}$ equations.

A brute force approach of iterating through all q^n possible assignments of x_i's may also be considered since the question asks to give a deterministic algorithm, not an efficient one.