INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR	
	Stamp / Signature of the Invigilator

EXAMINATION (End Semester)	SEMESTER (Autumn)

Roll Number										Section		Name	
Subject Number	C	S	6	0	0	8	6	Subject Name	*Selected Topics in Algorithms*				

Department / Center of the Student		Additional sheets	

Important Instructions and Guidelines for Students

1. You must occupy your seat as per the Examination Schedule/Sitting Plan.
2. Do not keep mobile phones or any similar electronic gadgets with you even in the switched off mode.
3. Loose papers, class notes, books or any such materials must not be in your possession, even if they are irrelevant to the subject you are taking examination.
4. Data book, codes, graph papers, relevant standard tables/charts or any other materials are allowed only when instructed by the paper-setter.
5. Use of instrument box, pencil box and non-programmable calculator is allowed during the examination. However, exchange of these items or any other papers (including question papers) is not permitted.
6. Write on both sides of the answer script and do not tear off any page. **Use last page(s) of the answer script for rough work.** Report to the invigilator if the answer script has torn or distorted page(s).
7. It is your responsibility to ensure that you have signed the Attendance Sheet. Keep your Admit Card/Identity Card on the desk for checking by the invigilator.
8. You may leave the examination hall for wash room or for drinking water for a very short period. Record your absence from the Examination Hall in the register provided. Smoking and the consumption of any kind of beverages are strictly prohibited inside the Examination Hall.
9. Do not leave the Examination Hall without submitting your answer script to the invigilator. **In any case, you are not allowed to take away the answer script with you.** After the completion of the examination, do not leave the seat until the invigilators collect all the answer scripts.
10. During the examination, either inside or outside the Examination Hall, gathering information from any kind of sources or exchanging information with others or any such attempt will be treated as **‘unfair means’**. Do not adopt unfair means and do not indulge in unseemly behavior.

Violation of any of the above instructions may lead to severe punishment.

Signature of the Student

To be filled in by the examiner

Question Number	1	2	3	4	5	6	7	8	9	10	Total
Marks Obtained											

Marks obtained (in words)	Signature of the Examiner	Signature of the Scrutineer

CS60035/CS60086 Selected Topics in Algorithms, Autumn 2016–2017

End-Semester Test

28–November–2016 | CSE-107, 2:00–5:00pm | Maximum marks: 70

[*Write your answers in the question paper itself. Be brief and precise. Answer <u>all</u> questions.*]

1. Let $\Sigma = \{a,b,c\}$ be an alphabet, and n a positive integer. Σ^n is the set of all strings over Σ of length n. Denote by $\mathrm{NOREP}_n \subseteq \Sigma^n$ the set of all strings over Σ of length n, in which no symbol is repeated in any two consecutive positions.

(a) You choose a string α from Σ^n uniformly randomly. What is the probability that $\alpha \in \mathrm{NOREP}_n$? **(5)**

Solution Let $S(n) = |\mathrm{NOREP}_n|$. We have $S(1) = 3$ (there cannot be any repetition in a string of length one), and $S(n) = 2S(n-1)$ for $n \geqslant 2$ (for a string $x_1x_2\ldots x_n$ to be in NOREP_n, the prefix $x_1x_2\ldots x_{n-1}$ must be in NOREP_{n-1}, and x_n must be different from x_{n-1}). The recurrence can be unfolded as $S(n) = 2S(n-1) = 2^2S(n-2) = \cdots = 2^{n-1}S(1) = 2^{n-1} \times 3$. Therefore the desired probability is

$$\frac{|\mathrm{NOREP}_n|}{|\Sigma^n|} = \frac{2^{n-1} \times 3}{3^n} = \left(\frac{2}{3}\right)^{n-1}.$$

(b) Suppose that we keep on choosing $\alpha \in_U \Sigma^n$, and stop as soon as a string of NOREP_n is found. What is the expected running time of this algorithm? **(5)**

Solution The probability that t iterations are needed to get a string of NOREP_n is $\left(1 - \left(\frac{2}{3}\right)^{n-1}\right)^{t-1} \times \left(\frac{2}{3}\right)^{n-1}$. Therefore the expected number of iterations is

$$\mathrm{E}[t] = \left(\frac{2}{3}\right)^{n-1} \times \sum_{t \geqslant 1} t \left(1 - \left(\frac{2}{3}\right)^{n-1}\right)^{t-1} = \left(\frac{2}{3}\right)^{n-1} \times \left[\frac{1}{\left(1 - \left(1 - \left(\frac{2}{3}\right)^{n-1}\right)\right)^2}\right] = \left(\frac{3}{2}\right)^{n-1}.$$

The expected running time is therefore $O(n(3/2)^n)$, which is exponential in n.

(c) Design an efficient algorithm that outputs a uniformly random member of NOREP_n. **(5)**

Solution

1. Choose $x_1 \in_U \Sigma$.
2. For $i = 2,3,\ldots,n$, choose $x_i \in_U \Sigma \setminus \{x_{i-1}\}$.
3. Return $x_1x_2\ldots x_n$.

2. Let A, B, C be $n \times n$ matrices with elements from a field K. Your task is to check whether $AB = C$. You take an n-dimensional column vector $\mathbf{v}$ with each entry chosen uniformly randomly from $\{0, 1\}$. You then check whether the equality $AB\mathbf{v} = C\mathbf{v}$ holds, and output *yes* or *no* accordingly.

(a) Argue that this test can be implemented to run in $O(n^2)$ time. **(5)**

Solution An $n \times n$ matrix can be multiplied by an n-vector in $O(n^2)$ time. It suffices to compute three matrix vector products $\mathbf{w} = B\mathbf{v}$, $\mathbf{x} = A\mathbf{w}$, and $\mathbf{y} = C\mathbf{v}$. Finally, the vectors $\mathbf{x}$ and $\mathbf{y}$ can be compared in $O(n)$ time.

(b) What is the error probability associated with this test? **(5)**

Solution Let $D = AB$. If the test outputs *no*, we certainly have $D \neq C$. The only possible error is the situation when $D \neq C$, but the test outputs *yes*. Since $D \neq C$, these two matrices differ in at least one position. Let $d_{ij} \neq c_{ij}$. The equality of the i-th elements in the fingerprinting vectors $\mathbf{x}$ and $\mathbf{y}$ requires

$$x_i = \sum_{k=1}^{n} d_{ik} v_k = \sum_{k=1}^{n} c_{ik} v_k = y_i.$$

This, in turn, implies that

$$v_i = (d_{ij} - c_{ij})^{-1} \sum_{\substack{k=1 \\ k \neq j}}^{n} (c_{ik} - d_{ik}) v_k.$$

Since the elements of $\mathbf{v}$ are chosen uniformly randomly and independently of one another, the last equality is satisfied with probability at most $\frac{1}{2}$, that is, the error probability is $\leqslant \frac{1}{2}$.

3. Let n be a sufficiently large integer, and $f : \mathbb{Z}_n \to \mathbb{Z}_n$ a fixed function satisfying $f(x+y) = f(x) + f(y)$ for all $x, y \in \mathbb{Z}_n$, where the additions are modulo n. You are given a faulty blackbox for evaluating f. Upon the input of an $x \in \mathbb{Z}_n$, the blackbox produces an output $y \in \mathbb{Z}_n$. We have $y = f(x)$ with probability $\frac{9}{10}$. If $y \neq f(x)$ (this happens with probability $\frac{1}{10}$), the output y is a predetermined randomly chosen element of $\mathbb{Z}_n$. The blackbox is deterministic in the sense that the output y does not change in different invocations of the blackbox, so long as the input x remains fixed. That is, for any given x, the output is always the same correct or faulty value of $f(x)$. Your task is to compute the value of $f(z)$ for some $z \in \mathbb{Z}_n$ supplied to you. Assume that each invocation of the blackbox takes one unit time.

(a) Propose an efficient randomized algorithm for solving your problem. Your algorithm should have an error/failure probability of at most ε (some small positive value supplied to you). **(5)**

Solution

1. Choose t based on the error bound ε (see Part (b)).
2. For $i = 1, 2, \ldots, t$, repeat:
   - Choose $z_i \in_U \mathbb{Z}_n$.
   - Invoke the blackbox to get the values $b_i = f_{blackbox}(z + z_i)$ and $c_i = f_{blackbox}(z_i)$.
   - Store $A[i] = b_i - c_i \pmod n$.
3. If the array $A[1 \ldots t]$ contains a majority element m, return m, else return *failure*.

(b) Argue that your algorithm satisfies the bound ε on error/failure probability as given in Part (a). **(5)**

Solution There are two ways this algorithm may malfunction. First, it outputs a wrong element as $f(z)$. This means that A stores a majority element different from the correct $f(z)$. This, in turn, implies that more than $\lfloor t/2 \rfloor$ iterations of the for loop encounter a situation when either b_i or c_i is faulty. Moreover, these faulty pairs give the same value of $b_i - c_i$. But since these faulty values are uniformly distributed in $\mathbb{Z}_n$, the probability that there is a faulty majority element is at most $1/n^{\lfloor t/2 \rfloor}$. This is extremely low assuming that n is sufficiently large.

So we concentrate on the second drawback of the algorithm, that is, we consider the situation when A does not contain a majority element, and the algorithm outputs *failure*. Each iteration of the for loop is a Bernoulli trial with success probability 0.81, so the probability that A does not contain a majority element is

$$p_t = \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} (0.81)^i (0.19)^{t-i}.$$

Given ε, we can choose a value of t for which $p_t \leqslant \varepsilon$. For example, if $t = 100$, then $p_t \approx 3 \times 10^{-12}$.

Solution (*continued*) If we desire a closed-form expression for t, we may proceed as follows.

$$\begin{aligned}
p_t &\leqslant \left(\sum_{i=0}^{\lfloor t/2\rfloor}\binom{t}{i}\right)\left(\sum_{i=0}^{\lfloor t/2\rfloor}(0.81)^i(0.19)^{t-i}\right) \leqslant 2^{t-1}\times(0.81)^t\times\sum_{i=0}^{\lfloor t/2\rfloor}\left(\frac{0.19}{0.81}\right)^{t-i} \\
&= 2^{t-1}\times(0.81)^t\times\sum_{j=\lceil t/2\rceil}^{t}\left(\frac{0.19}{0.81}\right)^{j} \leqslant 2^{t-1}\times(0.81)^t\times\left(\frac{0.19}{0.81}\right)^{\lceil t/2\rceil}\times\frac{1}{1-\dfrac{0.19}{0.81}} \\
&\leqslant 2^{t-1}\times(0.81)^t\times\frac{0.81}{0.62}\times\left(\frac{0.19}{0.81}\right)^{(t+1)/2} = \left(\frac{\sqrt{0.19\times0.81}}{2\times0.62}\right)\times(4\times0.19\times0.81)^{t/2} \\
&\leqslant 0.3164\times(0.6156)^{t/2} \leqslant \varepsilon
\end{aligned}$$

implies that we can take

$$t = \left\lceil \frac{2\times\Big(\log(1/\varepsilon)-\log 3.161\Big)}{\log 1.6244}\right\rceil .$$

(c) What is the running time of your algorithm? **(5)**

Solution Each iteration of the for loop takes $O(1)$ time. We need $t = O(\log(1/\varepsilon))$ iterations. So the running time if $O(\log(1/\varepsilon))$. In particular, if ε is constant, the running time is $O(1)$.

4. Let t be a positive integer. We choose t bits $b_1, b_2, \ldots, b_t$ uniformly randomly and independently of one another. For any non-empty subset I of $\{1,2,\ldots,t\}$, define the bit $X_I = \bigoplus_{i \in I} b_i$. Consider the $2^t - 1$ random variables X_I corresponding to all non-empty subsets I of $\{1,2,\ldots,t\}$.

(a) Prove that the $2^t - 1$ random variables X_I are pairwise independent. **(5)**

Solution Let I be a subset of $\{1,2,\ldots,t\}$ of size r. Since $\binom{r}{0}+\binom{r}{2}+\binom{r}{4}+\cdots = 2^{r-1} = \binom{r}{1}+\binom{r}{3}+\binom{r}{5}+\cdots$, we conclude that $\Pr[X_I = 0] = \Pr[X_I = 1] = \frac{2^{r-1}}{2^r} = \frac{1}{2}$. Since the bits $b_1, b_2, \ldots, b_t$ are chosen independently, for k pairwise disjoint non-empty subsets $I_1, I_2, \ldots, I_k$ of $\{1,2,\ldots,t\}$, we have $\Pr[I_1 = \alpha_1, I_2 = \alpha_2, \ldots, I_k = \alpha_k] = \Pr[I_1 = \alpha_1]\Pr[I_2 = \alpha_2]\cdots\Pr[I_k = \alpha_k] = \frac{1}{2^k}$ for all possible bit values $\alpha_1, \alpha_2, \ldots, \alpha_k$. Now, take two different non-empty subsets I, J of $\{1,2,\ldots,t\}$. We need to show that $\Pr[X_I = \alpha, X_J = \beta] = \frac{1}{4}$. We show it only for $\alpha = \beta = 0$, the argument for other three cases being very similar. If I and J are disjoint, we have already proved the result, so suppose that $I \cap J \neq \emptyset$. If $I \subseteq J$, then $X_I = X_J = 0$ if and only if $X_I = 0$ and $X_{J \setminus I} = 0$. But I and $J \setminus I$ are disjoint and non-empty, so $\Pr[X_I = X_J = 0] = \frac{1}{4}$. The case $J \subseteq I$ can be analogously handled. Finally, suppose that $I \not\subseteq J$ and $J \not\subseteq I$. In this case, $X_I = X_J = 0$ if and only if either $X_{I \setminus J} = X_{I \cap J} = X_{J \setminus I} = 0$ or $X_{I \setminus J} = X_{I \cap J} = X_{J \setminus I} = 1$. But the sets $I \setminus J$, $I \cap J$, and $J \setminus I$ are pairwise disjoint, so in this case, we have $\Pr[X_I = X_J = 0] = \frac{1}{8} + \frac{1}{8} = \frac{1}{4}$.

(b) Prove that the $2^t - 1$ random variables X_I are not three-wise independent. **(5)**

Solution Take two disjoint non-empty subsets I, J of $\{1,2,\ldots,t\}$, and set $K = I \cup J$. We have $X_K = X_I \oplus X_J$. This implies, for example, that $\Pr[X_I = X_J = X_K = 0] = \Pr[X_I = X_J = 0] = \frac{1}{4}$, and $\Pr[X_I = X_J = 0, X_K = 1] = 0$.

(c) Argue why these $2^t - 1$ bits X_I cannot be used as such to derandomize the random sampling algorithm for MAX_3SAT. **(5)**

Solution For derandomizing the random sampling algorithm for MAX_3SAT, we need n three-wise independent bits (where n is the number of variables). Part (b) implies that the bits X_I cannot be straightaway used as these n variables.

5. Consider the random sampling algorithm for the MAX_CUT problem in an undirected graph $G = (V, E)$. The algorithm produces a cut (S, T) for which each vertex $v_i \in V$ is put in S or T with probability $\frac{1}{2}$ and independently of one another. The objective is to maximize the count of edges with one endpoint in S and the other in T. This leads to a randomized 2-expected approximation algorithm. Let $V = \{v_1, v_2, \ldots, v_n\}$. Denote by $X_1, X_2, \ldots, X_n$ the random variables standing for the n choices for the vertices, with $X_i = 0$ (resp. 1) meaning that the vertex v_i goes to S (resp. T).

(a) Argue that the expected approximation ratio remains $\leqslant 2$ if the variables $X_1, X_2, \ldots, X_n$ are pairwise independent, that is, if for different i, j and for $\alpha, \beta \in \{0, 1\}$, we have $\text{Prob}[X_i = \alpha, X_j = \beta] = \frac{1}{4}$. **(5)**

Solution Let $m = |E|$, and Z_j for $j = 1, 2, \ldots, m$ the indicator variable that the j-th edge e_j is a cut edge. Then $Z = \sum_{j=1}^{m} Z_j$ is the number of cut edges. By linearity of expectation, we have

$$\mathrm{E}[Z] = \mathrm{E}\left[\sum_{j=1}^{m} Z_j\right] = \sum_{j=1}^{m} \mathrm{E}[Z_j].$$

Now, $e_j = (v_{i_1}, v_{i_2})$ if and only if either $X_{i_1} = 0, X_{i_2} = 1$ or $X_{i_1} = 1, X_{i_2} = 0$. If the variables X_i are pairwise independent, we have

$$\mathrm{E}[Z_j] = \Pr[X_{i_1} = 0] \Pr[X_{i_2} = 1] + \Pr[X_{i_1} = 1] \Pr[X_{i_2} = 0] = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}.$$

Therefore $\mathrm{E}[Z] = m/2$, and the maximum possible cut size is m.

(b) Using the bits X_I of Exercise 4, derandomize the random sampling algorithm for MAX_CUT. Mention clearly the choice of t, the steps of the derandomized algorithm, and why this is a deterministic polynomial-time 2-approximation algorithm for MAX_CUT. Do not use the polynomial-based construction discussed in the class. **(10)**

Solution

1. Choose $t = \lceil \log_2(n+1) \rceil$ (where $n = |V|$).
2. There are $2^t - 1 \geqslant n$ non-empty subsets of $\{1, 2, \ldots, t\}$. Pick any collection $C = (I_1, I_2, \ldots, I_n)$ of n distinct non-empty subsets of $\{1, 2, \ldots, t\}$.
3. For each of the 2^t choices for t bits $b_1, b_2, \ldots, b_t$, repeat:
   - Set $S = T = \emptyset$.
   - For $i = 1, 2, \ldots, n$, compute $X_i = \bigoplus_{k \in I_i} b_k$; and if $X_i = 0$, put v_i in S, else put v_i in T.
   - If (S, T) has size larger than the best known cut discovered so far, remember (S, T).
4. Return the best cut (S, T) obtained above.

The running time of this algorithm is $\mathrm{O}(2^t(nt + m))$. By the choice of t, we have $n \leqslant 2^{t+1}$, that is, this running time is $\mathrm{O}(n(n \log n + m))$.

Since the bits X_I (and so the bits X_i) are pairwise independent, we have $\mathrm{E}[Z] = m/2$. But then, for at least one choice of $b_1, b_2, \ldots, b_t$, we have $|c(S, T)| \geqslant m/2$. Since all the choices of the t bits are exhaustively looked into, this cut must be detected by the algorithm (in some iteration of Step 3).

Use this space for leftover answers and rough work

Use this space for leftover answers and rough work

Use this space for leftover answers and rough work