CS761 Derandomization and Pseudorandomness **July-Nov 2022**

Assignment 1

Total Marks: 60 *Deadline: Oct 2, 23:59*

Note: The assignment needs to be done individually. Any kind of discussion with other students is not allowed. Avoid searching for the problem statement on the internet. If you feel the need to discuss anything, you can discuss with the instructor. You can directly use any result proved in the class.

Que 1 [20 marks]. Consider a computational problem where the input is divided between two devices, say as $x = (x_1, x_2)$. The two devices follow the below protocol P.

- The first device gets an input x_1, does a randomized computation using r random bits (say denoted by s_1).
- After the computation, the first device sends k bits of information to the second device.
- The second device then does another randomized computation on input x_2 and the k bits provided by the first device. For simplicity, suppose the second device also uses r random bits (say denoted by s_2). After the computation, the second device outputs the answer, yes or no.

Assume that the output of the protocol $P(x, s_1, s_2)$ is the correct answer for the problem at hand with a good probability (probability over choice of (s_1, s_2)).

We want to derandomize this computation protocol. Observe that when there is zero bits of communication, then we could provide the same random string $s_1 = s_2$ to both the devices and it will not make a difference in the probability of correctness. This is simply because what device one is computing is irrelevant. Intuitively, when k is much smaller than r, we may be able to reuse the most of the first r random bits for the second device. But, it's not clear which of the bits. It depends on exactly what information is being sent. Our goal is to construct pseudorandom bits that will be guaranteed to work without any assumption on what computation is being done and what information is being sent.

Design a function $\mathcal{G}$ that takes $r + \ell$ bits as input, where $\ell = O(k)$ and outputs $2r$ bits such that any such two device computation protocol will behave (almost) the same way whether you give $2r$ truly random bits or the output of $\mathcal{G}$. More formally, we want that for any two device computation protocol P,

$$\left| \Pr_{(s_1,s_2)\sim U_{2r}}[P(x, s_1, s_2) = \text{yes}] - \Pr_{(s_1,s_2)\sim \mathcal{G}(U_{r+\ell})}[P(x, s_1, s_2) = \text{yes}] \right| \leq 1/2^k.$$

Hint: Suppose G is a Ramanujan expander graph with 2^r vertices, degree 2^ℓ and $\lambda \approx 2/\sqrt{2^\ell} = 1/2^{\ell-1}$. Now define the function $\mathcal{G}$ on input (u, i) to output (u, v), where v is the ith neighbor of vertex u. Note that $1 \leq u, v \leq 2^k$ and $1 \leq i \leq 2^\ell$.

To prove the desired bound, the expander mixing lemma will be useful. Let $c_1, c_2, \ldots, c_{2^k}$ be the possible values of k bits sent by device one. Let S_j be the set of random strings on which the device one sends c_j to device two. Upon receiving c_j, let T_j be set of random strings for device two which lead to the correct answer. Try to compute the two probabilities (under random and pseudorandom), for each fixed choice c_j and then take a sum. You will need to apply expander mixing lemma for (S_j, T_j).

Que 2 [5 marks]. Prove that to generate n 4-wise independent bits, one needs at least $2 \log n$ (independent) random bits.

Hint: Using a some 4-wise independent bits, can you generate a larger number of pairwise independent bits? If yes, then you can try to apply the lower bound for pairwise independence.

Que 3 [10 marks]. Let $\mathbb{F}$ be a finite field. Consider a graph with vertex set $\mathbb{F}^2$ and edge set $\{((a, b), (c, d)) : ac = b + d\}$. That is, we connect point (a, b) to all points on the line $y = ax - b$. Prove that G is $\mathbb{F}$-regular and $\lambda(G) \leq 1/\sqrt{|\mathbb{F}|}$.

Hint: Consider G^2. Write its random walk matrix as $J + (1/|\mathbb{F}|)E$, where J is the random walk matrix for the complete graph (with self loops). Bound the largest absolute eigenvalue of E by 1.

Que 4 [10 marks]. Suppose there are two parties who are supposed to have their own copies of some data. They want to verify whether their copies are actually same. A trivial way is to send the whole data, but we want to do something more efficient. Suppose the data is n-bit long; the first party has $a = a_1 a_2 \cdots a_n$ and the second party has $b = b_1 b_2 \cdots b_n$. Let us define two matrices

$$M_0 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \text{ and } M_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

The first party chooses a random integer p from $\{2, 3, \ldots, 2n\}$. Then it sends the following matrix product to the second party after taking modulo p, along with the number p.

$$M_a = M_{a_1} M_{a_2} \cdots M_{a_n} \pmod{p}.$$

The second party computes the same product for b

$$M_b = M_{b_1} M_{b_2} \cdots M_{b_n} \pmod{p}$$

and verifies whether $M_a \equiv M_b \pmod{p}$.

Prove that the algorithm works correctly with a good probability.

Hint: You can assume that the lcm of first n numbers is at least 2^n. You may want to argue that the map $a \mapsto M_a$ (without mod) is one-one. For this you can show that a can be recovered from M_a; look at M_a and first try to find the first bit a_1.

Que 5 [5+10 marks]. Consider another algorithm for estimating number of distinct items in a stream. For a fixed number k, we try to predict whether the number of distinct items is at least $2^{\ell+2}$ or at most $2^{\ell-2}$. Running this procedure for different values of k will give us an approximate value of number of distinct element. Let the items come from the universe $[N] = \{1, 2, \ldots, N\}$.

- $h\colon [N] \to \{0,1\}^\ell$ is taken to be pairwise independent random function. That is, for any distinct $i, j \in [N]$ and any $\alpha, \beta \in \{0,1\}^\ell$, we have

$$\Pr[h(i) = \alpha \mid h(j) = \beta] = 1/2^\ell.$$

- Initialize Flag := False.
- For each item a in the stream:

  if $h(a) = 0^\ell$ then set Flag := True.
- If Flag = True then output yes otherwise no.

Prove that

- if the number of distinct items is at most $2^{\ell-2}$, then the algorithm says no with probability at least 3/4.
- if the number of distinct items is at least $2^{\ell+2}$, then the algorithm says yes with probability at least 3/4.

Hint: the analysis is along similar lines as what we did in the class, but simpler.