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

Assignment 2

Total Marks: 60 *Deadline: Nov 21, 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 [15 marks]. We want to give a counting (probabilistic) argument that there exist linear codes over alphabet $\{0, 1\}$ with constant rate r and constant distance δ.

$GF(2)$ is the field of size 2. Consider a randomly constructed $n \times k$ matrix E (each entry is uniformly independently chosen from $GF(2)$). Consider the code

$$\{Ex : x \in GF(2)^k\}.$$

Show that for an appropriate choice of k, the above code has rate at least r and distance at least δ with nonzero probability. You can choose r and δ as some constants between 0 and 1, as convenient. You can directly use Chernoff bounds, if needed.

Que 2 [10 marks]. Using counting/probabilistic arguments, show that for any large enough n, there exists a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ such that

$$H_{avg}(f) \geq 2^{n/10}.$$

You can directly use Chernoff bounds, if needed.

Que 3 [15 marks]. Prove a generalized version of Isolation Lemma. That is, let $\mathcal{S} \subseteq \{0, 1, 2, \dots, d\}^m$ be a set of vectors. Let us choose a (uniformly) random vector $w \in \{0, 1, 2, \dots, 2dm\}^m$. Show that with a good probability there is a unique element x in $\mathcal{S}$ that minimizes $w \cdot x$.

Que 4 [5 marks]. Let $\mathbb{F}$ be finite field. Let $E \in \mathbb{F}^{n\times k}$ be a matrix with the property that every square submatrix is invertible. Prove that the distance of the following code (over alphabet $\mathbb{F}$) is $(n - k + 1)/n$.

$$\{Ex : x \in \mathbb{F}^k\}.$$

Que 5 [15 marks]. Recall the bipartite matching algorithm we discussed in the class. We showed a construction of a quasi-polynomially large family of weight assignments where weights were quasi-polynomially bounded and the guarantee was that for any bipartite graph, one of the weight assignments in the family will be isolating. This gave us a deterministic parallel algorithm for bipartite matching, which can use quasi-polynomially many processors and poly-logarithmic time.

Now, use the same construction ideas to solve a related question where we avoid using quasi-polynomially many processors. Let FIND be the problem of finding a perfect matching in a given bipartite graph. Let OPT be the problem of finding the weight of the minimum weight perfect matching in a given bipartite graph with small (polynomially bounded) edge weights. Give a deterministic parallel reduction from FIND to OPT that uses only polynomially many processors. In other words, assume you have an oracle for OPT and using that oracle you have to find a perfect matching. Note that OPT oracle only gives the **weight** of minimum weight perfect matching, and not the perfect matching.