CSE 632: Analysis of Algorithms II Fall 2017

# Homework 4

*Lecturer: Shi Li* **Deadline: 12/6/2017**

**Instructions and Policy.** Each student should write their own solutions independently and needs to indicate the names of the people you discussed a problem with. It is highly recommended that you think about each problem on your own for enough time before you discuss with others. It is recommended that you type your solutions using latex and submit the pdf file via UBLearns.

**Problem 0 (0 points)** You do not need to write down the proof for this problem; just verify it. Let $p > 4$ be a prime number and let $\mathbb{Z}_p = \{0, 1, 2, \cdots, p-1\}$. For every $a, b, c, d, x \in \mathbb{Z}_p$, let $f_{a,b,c,d}(x) = ax^3 + bx^2 + cx + d \bmod p$. Show that for any 4 distinct numbers $x_1, x_2, x_3, x_4 \in \mathbb{Z}_p$ and 4 numbers $y_1, y_2, y_3, y_4 \in \mathbb{Z}_p$, there is exactly one tuple $(a, b, c, d) \in \mathbb{Z}_p^4$, such that $f_{a,b,c,d}(x_1) = y_1, f_{a,b,c,d}(x_2) = y_2, f_{a,b,c,d}(x_3) = y_3$ and $f_{a,b,c,d}(x_4) = y_4$.

**Problem 1 (30 points)** In the maximum 2-SAT problem, we are given $n$ boolean variables $x_1, x_2, \cdots, x_n$ and $m$ clauses, where each clause is the disjunction (the "or" operation) of 2 literals (a literal is either $x_i$ or $\neg x_i$). The goal of the problem is to find an assignment to $x_1, x_2, \cdots, x_n$ so as to maximize the number of satisfied clauses. Use SDP to design a 0.878-approximation algorithm for this problem.

Hint: in the vector programming, let $y_0$ be a unit-length vector that stands for "true"; then $-y_0$ stands for "false". Use $y_0, y_i$ and $y_j$ to express whether a clause involving $x_i$ and $x_j$ is correct or not.

**Problem 2 (40 points)** Recall that in the weighted set cover problem, we are given $m$ subsets $S_1, S_2, \cdots, S_m$ of $[n]$, and a weight vector $w = (w_1, w_2, \cdots, w_m) \in \mathbb{R}^m_{\geq 0}$. The goal is to choose a set $I \subseteq [m]$ such that $\bigcup_{i \in I} S_i = [n]$ so as to minimize $\sum_{i \in I} w_i$. Consider the following natural LP relaxation for the problem:

$$\min \quad \sum_{i \in I} w_i x_i \qquad \text{s.t.}$$

$$\sum_{i: j \in S_i} x_i \geq 1 \quad \forall j \in [n], \qquad\qquad x_i \geq 0 \quad \forall i \in [m]$$

Assume we have an efficient rounding algorithm $\mathcal{A}$, that given any weighted set cover instance $(n, m, \{S_i\}_{i \in m}, w)$, and any valid solution $x$ to the above linear programming, outputs a valid solution $I \subseteq [m]$ to the instance such that $\sum_{i \in I} w_i \leq \alpha_n \sum_{i \in [m]} w_i x_i$, for some function $\alpha : \mathbb{Z}_{>0} \to [1, \infty)$.

Use $\mathcal{A}$ as a black box to design an efficient randomized rounding algorithm $\mathcal{A}'$ that is "oblivious" to the weight vector $w$. More specifically, let $\epsilon > 0$ be a given parameter.

Then given $(n, m, \{S_i\}_{i \in m})$ and a valid solution $x$ to the LP, $\mathcal{A}'$ should randomly output a valid solution $I$ to the instance such that $\Pr[i \in I] \leq (1+\epsilon)\alpha_n x_i$ for every $i \in [n]$. Notice that $w$ is not given to $\mathcal{A}'$; also notice that whether $x$ or $I$ is valid is independent of the $w$ vector; so the requirement for $\mathcal{A}'$ is well-defined.

Hint: design a 0-sum game between a $w$-player and an $I$-player; then use the multiplicative weight update method to find the strategy for the $I$-player.

**Problem 3 (30 points)** Assume there is a data stream $i_1, i_2, i_3, \cdots, i_n$ where each element $i_t$ is in $[m]$. Assume we know an upper bound $N$ of $n$ (but we do not know $n$). Design an one-pass streaming algorithm that finds an $\epsilon$-approximate median of the multi-set $\{i_1, i_2, i_3, \cdots, i_n\}$: a number $a$ such that

$$|\{t \in [n] : i_t < a\}| \leq (1+\epsilon)n/2 \quad \text{and} \quad |\{t \in [n] : i_t > a\}| \leq (1+\epsilon)n/2.$$

Your streaming algorithm should use $\text{poly}(1/\epsilon, \log N, \log m)$ space and succeed with probability at least $2/3$.

Hint: show how to uniformly sample $k$ elements from the multi-set $\{i_1, i_2, \cdots, i_n\}$. Then, use Chernoff bound to show that if $k$ is large enough, then outputting the median of the $k$ sampled elements solves the problem.