CS331: Algorithms and Complexity
Homework IV

Trung Dang Ryan Park Kevin Tian

Due date: November 4, 2024, end of day (11:59 PM), uploaded to Canvas.

Late policy: 15% off if submitted late, and 15% off for every further 24 hours before submission.

Please list all collaborators on the first page of your solutions.

When runtimes are unspecified, slower runtimes than the intended solution receive partial credit.

1 Problem 1

Prove either Eq. (12) (i.e., Nash equilibria in zero-sum games) or Eq. (14) (i.e., the strong maxflow-mincut theorem) of Part VI of the notes, reproduced below, via LP duality. In particular, assuming both Eq. (8) and Eq. (9) of Part VI of the notes are true, prove *one* of the following.

(i) **(20 points)** Let $\mathbf{A} \in \mathbb{R}^{n \times d}$. Then,

$$\max_{\substack{\mathbf{y} \in \mathbb{R}^n_{\geq 0} \\ \sum_{i \in [n]} \mathbf{y}_i = 1}} \min_{j \in [d]} \left[\mathbf{A}^\top \mathbf{y}\right]_j = \min_{\substack{\mathbf{x} \in \mathbb{R}^d_{\geq 0} \\ \sum_{j \in [d]} \mathbf{x}_j = 1}} \max_{i \in [n]} \left[\mathbf{A}\mathbf{x}\right]_i.$$

(ii) **(20 points)** Let $\mathbf{A} \in \{-1, 0, 1\}^{V \times E}$ be the incidence matrix of $G = (V, E, \mathbf{w})$, defined in Eq. (13), Part VI of the notes, let $U := V \setminus \{s, t\}$ for $s, t \in V$ with $s \neq t$, and let $\mathbf{c} \in \mathbb{R}^E_{\geq 0}$. Then,

$$\max_{\substack{\mathbf{x} \in \mathbb{R}^E_{\geq 0} \\ \mathbf{x} \leq \mathbf{c} \\ [\mathbf{A}\mathbf{x}]_U = \mathbf{0}_U}} [\mathbf{A}\mathbf{x}]_s = \min_{\substack{\mathbf{y} \in \mathbb{R}^E_{\geq 0}, \mathbf{z} \in \mathbb{R}^V \\ \mathbf{z}_s = 1, \mathbf{z}_t = 0 \\ \mathbf{y}_e \geq \mathbf{z}_u - \mathbf{z}_v \text{ for all } e = (u,v) \in E}} \mathbf{c}^\top \mathbf{y}.$$

No algorithm is necessary to solve this problem.

2 Problem 2

(20 points) For some $L \geq 2\epsilon > 0$, let $f : [0, 1] \to \mathbb{R}$ be L-Lipschitz and strictly convex. Give an algorithm that returns a point $\hat{x} \in [0, 1]$ such that $f(\hat{x}) - \min_{x \in [0,1]} f(x) \leq \epsilon$ using $O(\log(\frac{L}{\epsilon}))$ queries to f, and $O(\log(\frac{L}{\epsilon}))$ additional time. Your algorithm should be based on *ternary search*.

3 Problem 3

Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be a symmetric matrix. Prove the following bidirectional statements about $\mathbf{A}$.

(i) **(10 points)** $\mathbf{A}$ is positive semidefinite iff $\mathbf{A}_{S \times S}$ is positive semidefinite, for all $S \subseteq [d]$, where $\mathbf{A}_{S \times S}$ denotes the $|S| \times |S|$ submatrix formed by the entries $\{\mathbf{A}_{ij}\}_{(i,j) \in S \times S}$.

(ii) **(10 points)** $\mathbf{A}$ is positive semidefinite iff it can be written in the form $\mathbf{A} = \mathbf{B}^\top \mathbf{B}$, for some (potentially asymmetric) matrix $\mathbf{B} \in \mathbb{R}^{d \times d}$.

No algorithm is necessary to solve this problem.

4 Problem 4

Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ be given as Array instances, and assume $\mathbf{x}$ has monotonically increasing coordinates, i.e., $\mathbf{x}_1 < \mathbf{x}_2 < \ldots < \mathbf{x}_d$. For $k \in \mathbb{N}$, we say a function $f : \bigcup_{j\in[k]} I_j \to \mathbb{R}$ is *k-piecewise-constant* if, for some k non-overlapping contiguous intervals $\{I_j\}_{j\in[k]}$ that cover all the coordinates $\{\mathbf{x}_i\}_{i\in[d]}$, f is constant on each interval I_j for all $j \in [k]$, i.e., $f(x) = c_j$ for some $c_j \in \mathbb{R}$, for all $x \in I_j$. Each interval I_j should have both its left and right endpoints equal to coordinates of $\mathbf{x}$.

For example, if $d = 6$ and $k = 3$, one valid choice of a k-piecewise-constant function f is $f(x) = 3$ for $x \in I_1 := [\mathbf{x}_1, \mathbf{x}_2]$, $f(x) = -4$ for $x \in I_2 := [\mathbf{x}_3, \mathbf{x}_4]$, and $f(x) = 7$ for $x \in I_3 := [\mathbf{x}_5, \mathbf{x}_6]$.

(i) **(5 points)** Give an algorithm that on inputs $\mathbf{x}, \mathbf{y}$, returns the constant c_1 defining a 1-piecewise-constant $f : [\mathbf{x}_1, \mathbf{x}_d] \to \mathbb{R}$ that minimizes, over all 1-piecewise-constant functions,

$$\sum_{i\in[d]} (f(\mathbf{x}_i) - \mathbf{y}_i)^2 .$$

For full credit, your algorithm should run in time $O(d)$.

(ii) **(15 points)** Give an algorithm that on inputs $\mathbf{x}, \mathbf{y}$, and $C > 0$, returns the constants $\{c_j\}_{j\in[k]}$ defining a k-piecewise-constant $f : \bigcup_{j\in[k]} I_j \to \mathbb{R}$ that minimizes, over all k-piecewise-constant functions, and over all possible values of $k \in [d]$,

$$\sum_{i\in[d]} (f(\mathbf{x}_i) - \mathbf{y}_i)^2 + Ck.$$

The input $C > 0$ should be interpreted as the additional cost charged per piece used in defining f. For full credit, your algorithm should run in time $O(d^3)$.

5 Problem 5

(20 points) Complete the assignment at this link. This link is only accessible on your UT email.

You do not need to analyze any algorithms or runtimes for this problem. Please briefly describe your approach for each part, and follow the additional instructions for the last part.