

CS331: Algorithms and Complexity

Homework IV

Trung Dang

Nathan Mardanov

Kevin Tian

Due date: November 3, 2025, 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.

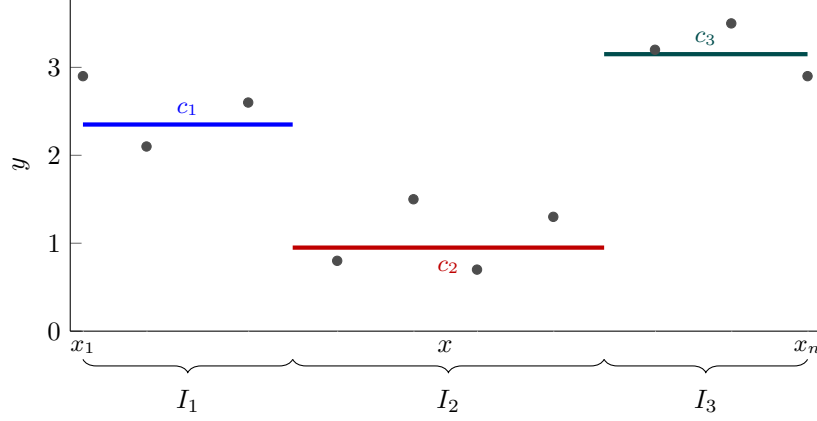
1 Problem 1

Let $f : [0, 1] \rightarrow \mathbb{R}$ be L -Lipschitz and strictly convex, and let $\epsilon > 0$.

- (i) **(5 points)** Suppose that $\epsilon \geq \frac{L}{2}$. 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 no queries to f , and $O(1)$ additional time.
- (ii) **(15 points)** Suppose that $\epsilon < \frac{L}{2}$. 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*.

2 Problem 2

Let $\{x_i \in \mathbb{R}\}_{i \in [n]}, \{y_i \in \mathbb{R}\}_{i \in [n]}$ be given as length- n **Array** instances, and assume $x_1 < x_2 < \dots < x_n$. For $k \in \mathbb{N}$, we say a function $f : [x_1, x_n] \rightarrow \mathbb{R}$ is *k-piecewise-constant* (k -PC) if, for some k non-overlapping intervals $\{I_j\}_{j \in [k]}$ that partition $[x_1, x_n]$, f is constant on each interval, i.e., for each $j \in [k]$, $f(x) = c_j$ for some $c_j \in \mathbb{R}$ and all $x \in I_j$.



- (i) **(5 points)** Give an algorithm that on inputs $\{x_i\}_{i \in [n]}, \{y_i\}_{i \in [n]}$, returns the c_1 defining a 1-PC $f : [x_1, x_n] \rightarrow \mathbb{R}$ that minimizes, over all 1-PC f ,

$$\sum_{i \in [n]} (f(x_i) - y_i)^2. \quad (1)$$

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

- (ii) **(15 points)** Give an algorithm that on inputs $\{x_i\}_{i \in [n]}, \{y_i\}_{i \in [n]}$, and $C > 0$, returns the $\{c_j\}_{j \in [k]}$ and $\{I_j\}_{j \in [k]}$ defining a k -PC $f : [x_1, x_n] \rightarrow \mathbb{R}$ that minimizes, over all k -PC f , and all possible $k \in [n]$,

$$\sum_{i \in [n]} (f(x_i) - y_i)^2 + Ck. \quad (2)$$

The input $C > 0$ should be interpreted as the additional cost charged per piece used to define f (without this cost, we would always choose $k = n$ to minimize (2)).

For full credit, your algorithm should run in time $O(n^3)$.

Note that although adjacent intervals I_j, I_{j+1} overlap at one endpoint, we can arbitrarily define f at this endpoint, by enforcing that all overlapping intervals end between adjacent x_i, x_{i+1} . This is because (1), (2) only depend on values of f at the $\{x_i\}_{i \in [n]}$.

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 $\{\mathbf{A}_{ij}\}_{(i,j) \in S \times S}$.

For example, below is an example of $\mathbf{A}$ and $\mathbf{A}_{S \times S}$ when $d = 3$ and $S = \{1, 2\}$:

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 3 \\ 1 & 3 & -2 \\ 3 & -2 & 4 \end{pmatrix}, \quad \mathbf{A}_{S \times S} = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$$

- (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 $G = (V, E)$ be a *bipartite graph*, i.e., $V = L \cup R$ and every edge $e = (u, v) \in E$ has $u \in L$ and $v \in R$. In the *vertex cover* problem, we want to assign values $x_v \geq 0$ to each vertex $v \in V$ minimizing the total value $\sum_{v \in V} x_v$, while ensuring that $x_u + x_v \geq 1$ for each edge $(u, v) \in E$.

When all $x_v \in \{0, 1\}$, we can interpret this problem as selecting the smallest possible set $S \subseteq V$ (i.e., the vertices v with $x_v = 1$) such that every edge has at least one endpoint in S .

- (i) **(10 points)** Write the vertex cover problem as an LP. Any of the forms of an LP that we covered in class is fine for this part.
- (ii) **(10 points)** Take the dual of the vertex cover LP. What problem (that we have already encountered) does it correspond to, when the decision variables are all restricted to be integers?

No algorithm is necessary to solve this problem.

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.