Final Exam, Advanced Algorithms 2019-2020

- You are only allowed to have a handwritten A4 page written on both sides.
- Communication, calculators, cell phones, computers, etc... are not allowed.
- Your explanations should be clear enough and in sufficient detail that a fellow student can understand them. In particular, do not only give pseudo-code without explanations. A good guideline is that a description of an algorithm should be such that a fellow student can easily implement the algorithm following the description.
- **You are allowed to refer to material covered in the lecture notes** including theorems without reproving them.
- **Problems are not necessarily ordered by difficulty.**
- **Do not touch until the start of the exam.**

Good luck!

Name: ______________________ **N° Sciper:** ______________

Problem 1 / 24 points	Problem 2 / 14 points	Problem 3 / 12 points	Problem 4 / 17 points	Problem 5 / 19 points	Problem 6 / 14 points

Total / 100

1 *(24 pts)* **Linear programming.**

1a *(12 pts)* **Duality.** Consider the following linear program:

$$\begin{array}{rl} \textbf{Minimize} & x_1 + 8x_2 + 20x_3 \\ \textbf{Subject to} & x_1 + 7x_2 + 2x_3 \geq 9 \\ & 3x_1 + \ \ x_2 + 4x_3 \geq 1 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Write down its dual and the complementarity slackness conditions.

Solution:

The dual problem of the abovementioned primal is the following:

$$\begin{array}{rl} \textbf{Maximize} & 9y_1 + y_2 \\ \textbf{Subject to} & y_1 + 3y_2 \leq 1 \\ & 7y_1 + \ \ y_2 \leq 8 \\ & 2y_1 + 4y_2 \leq 20 \\ & y_1, y_2 \geq 0 \end{array}$$

The complementarity slackness conditions say that if (x_1, x_2, x_3) and (y_1, y_2) are feasible solutions to the primal and the dual LPs respectivly, then :

$$\left.\begin{array}{rl} (x_1, x_2, x_3) & \text{is an optimal solution of the primal and} \\ (y_1, y_2) & \text{is an optimal solution of the dual} \end{array}\right\} \iff \left\{\begin{array}{lll} x_1 > 0 & \Rightarrow & y_1 + 3y_2 = 1 \\ x_2 > 0 & \Rightarrow & 7y_1 + \ \ y_2 = 8 \\ x_3 > 0 & \Rightarrow & 2y_1 + 4y_2 = 20 \\ y_1 > 0 & \Rightarrow & x_1 + 7x_2 + 2x_3 = 9 \\ y_2 > 0 & \Rightarrow & 3x_1 + \ \ x_2 + 4x_3 = 1 \end{array}\right.$$

1b *(12 pts)* **Simplex method.** Suppose we use the Simplex method to solve the following linear program:

$$\begin{aligned}
\textbf{Maximize} \quad & 6x_1 - 4x_2 - 9x_3 \\
\textbf{Subject to} \quad & x_1 - 2x_3 + s_1 = 5 \\
& 2x_1 + x_2 + 4x_3 + s_2 = 35 \\
& 4x_3 - 3x_2 + s_3 = 12 \\
& x_1,\ x_2,\ x_3,\ s_1,\ s_2,\ s_3 \geq 0
\end{aligned}$$

At the current step, we have the following Simplex tableau:

$$\begin{aligned}
x_1 &= 5 + 2x_3 - s_1 \\
s_2 &= 25 - x_2 - 8x_3 + 2s_1 \\
s_3 &= 12 + 3x_2 - 4x_3 \\
\hline
z &= 30 - 4x_2 + 3x_3 - 6s_1
\end{aligned}$$

Write the tableau obtained by executing one iteration (pivot) of the Simplex method starting from the above tableau.

Solution:

Only x_3 has a positive coefficient in z, we will pivot x_3. We have

$\nearrow x_3 \longrightarrow\ x_3 \leq\ \infty\ (1),\ x_3 \leq 25/8\ (2),\ x_3 \leq 12/4 = 3\ (3) \longrightarrow x_3 := 3,\ s_3 := 0$

$$\begin{aligned}
x_3 &= 3 + 3x_2/4 - s_3/4 \\
x_1 &= 5 + 2 \cdot (3 + 3x_2/4 - s_3/4) - s_1 \\
&= 11 + 3x_2/2 - s_3/2 - s_1 \\
s_2 &= 25 - x_2 - 8(3 - 3x_2/4 - s_3/4) + 2s_1 \\
&= 1 - 7x_2 + 2s_1 - 2s_3 \\
\hline
z &= 30 - 4x_2 + 3(3 + 3x_2/4 - s_3/4) - 6s_1 \\
&= 39 - 25x_2/4 - 6s_1 - 3s_3/4
\end{aligned}$$

$x_1 := 11 \quad x_2 := 0 \quad x_3 := 3 \quad s_1 := 0 \quad s_2 := 1 \quad s_3 := 0$

2 (14 pts) **Set Balancing.** You are given a collection of sets $S_1, S_2, \ldots, S_m$ in a universe U of size n with $|S_i| \leq t$ for all $i = 1, \ldots, m$. A function $\chi : U \to \{-1, +1\}$ is called a q-balanced coloring if for every $i = 1, \ldots, m$ one has

$$\left| \sum_{e \in S_i} \chi(e) \right| \leq q.$$

Give an efficient randomized algorithm for finding a q-balanced coloring with $q = O(\sqrt{t \log m})$. Your algorithm must succeed with probability at least $9/10$.

In this problem you should (a) design an efficient algorithm and (b) prove its correctness.

Solution:

Algorithm: sample $\chi(e) := r_e \sim \text{Uniform}(\{-1, 1\})$ for $j \in \{1, ..., n\}$.

Analysis: First we define event A as sampling a coloring which is not q-balanced. Then we can upper bound the probability if this event as:

$$\begin{aligned} \Pr[A] &= \Pr\left[\max_{i \in [m]} \left| \sum_{e \in S_i} r_e \right| \geq q\right] = \Pr\left[\left| \sum_{e \in S_1} r_e \right| \geq q \text{ OR } \ldots \text{ OR } \left| \sum_{e \in S_m} r_e \right| \geq q\right] \\ &\leq \sum_{i=1}^{m} \Pr\left[\left| \sum_{e \in S_i} r_e \right| \geq q\right] \leq \sum_{i=1}^{m} 2 \exp\left(-\frac{q^2}{2|S_i|}\right) \leq 2m \exp\left(-\frac{q^2}{2t}\right), \end{aligned}$$

where the inequalities follow from the union bound, Hoeffding's inequality and the fact that $|S_i| \leq t$ respectively.

Then by selecting $q = \sqrt{2t \log \frac{2m}{p}}$ for some failure probability p, we can achieve the desired bound $\Pr[A] \leq p$:

$$\Pr[A] \leq 2m \exp\left(-\frac{2t \log \frac{2m}{p}}{2t}\right) = 2m \frac{p}{2m} = p.$$

This shows in particular if the algorithm has to succeed with probability at least 0.9, having $q = \sqrt{2t \log \frac{2m}{0.1}} = O(\sqrt{t \log m})$ is sufficient.

3 *(12 pts)* **Alice, Bob and Charlie.** Suppose that Alice and Bob have two documents d_A and d_B respectively, and Charlie wants to learn about the difference between them. We represent each document by its word frequency vector as follows. We assume that words in d_A and d_B come from some dictionary of size n, and let $x \in \mathbb{R}^n$ be a vector such that for every word $i \in [n]$[1] the entry x_i equals the number of times the i-th word in the dictionary occurs in d_A. Similarly, let $y \in \mathbb{R}^n$ be a vector such that for every word $i \in [n]$ the entry y_i denotes the number of times the i-th word in the dictionary occurs in d_B. We assume that the number of words in each document is bounded by a polynomial in n.

In the subproblems below you will design efficient communication protocols for Alice and Bob that let Charlie learn about the difference of d_A and d_B. All protocols must succeed with probability at least 9/10. For all subproblems below you may assume that Alice, Bob and Charlie have a source of shared random bits.

3a (5 pts) Suppose that for an integer parameter k there exist at most k words that are present in d_A but not in d_B, and at most k words that are present in d_B but not in d_A. Show that Alice and Bob can each send a message of $O(k \log^2 n)$ bits to Charlie, from which Charlie can recover the words that are present in d_A but not in d_B.

3b (7 pts) Suppose that there exists $i^* \in [n]$ such that for all $i \in [n] \setminus \{i^*\}$ one has $|x_i - y_i| \leq 2$, and for i^* one has $|x_{i^*} - y_{i^*}| \geq n^{1/2}$. Show that Alice and Bob can each send a $O(\log^2 n)$-bit message to Charlie, from which Charlie can recover the identity of the special word i^*.

(Hint: recall one of the streaming algorithms covered in class, and note that it is a linear sketch that can be applied to arbitrary vectors of length n in with integer entries bounded by a polynomial in n.)

(Recall that you are allowed to refer to material covered in the lecture notes. In this problem you must explain how your protocol works and why it is correct with the required probability.)

Note: you do not need to do detailed calculations in this problem.

[1] We let $[n] := \{1, 2, \ldots, n\}$.

Solution:

Recall from lectures the CountSketch algorithm for ℓ_2 heavy hitters. Given a vector v, and a parameter $\phi \in (0,1)$, we can return an approximate vector $\widetilde{v}$ such that

$$\forall i: \; |v_i - \widetilde{v}_i| \leq \phi \|v\|_2/2.$$

This requires $O(\phi^{-2} \log^2 n)$ bits of space.

3a Define the indicator vectors $\overline{x}$ and $\overline{y}$ denoting whether each word of the dictionary appears at least once in d_A or d_B respectively. Formally $\overline{x}_i = \mathbf{1}(x_i > 0)$ and similarly $\overline{y}_i = \mathbf{1}(y_i > 1)$. Let Alice and Bob apply CountSketch to $\overline{x}$ and $\overline{y}$ respectively, with $\phi = \sqrt{1/3k}$ and with the same random seed, and send it to Charlie. This requires $O(k \log^2 n)$ space and allows Charlie to decode a CountSketch of $v = \overline{x} - \overline{y}$ where each coordinate is approximated with error $\sqrt{1/3k} \cdot \|v\|_2/2 < 1/2$. Charlie can simply recover the words in d_A but not d_B by looking at coordinates of $\widetilde{v}$ that are greater than $1/2$.

3b This time Alice and Bob apply CountSketch to x and y respectively, with $\phi = 1/3$ and with the same random seed, and send it to Charlie. This allows Charlie to decode a CountSketch of $v = x - y$. This takes $O(\log^2 n)$ and allows Charlie to recover any coordinate i such that $|v_i| > 1/9 \cdot \|v\|_2$. i^* is such a coordniate, since $\|v\|_2^2 \leq 4n + v_{i^*}^2$ and $v_{i^*}^2 \geq n$.

(This page is intentionally left blank.)

4 *(17 pts)* **Spectral graph theory.** Let $G = (V, E)$ be a d-regular undirected graph, and let $M = \frac{1}{d}A$ denote its normalized adjacency matrix. Let $1 = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-1} \geq \lambda_n$ denote the eigenvalues of M.

4a *(12 pts)* Prove that if $\lambda_{n-1} = -1$, then G is disconnected.

Solution: Suppose for contradiction that G is connected and $\lambda_{n-1} = -1$.

Claim 1 *If v is such that $Mv = -v$, then for some $a \in \mathbb{R}$ all coordinates of v are $\pm a$. Also, all edges of E go between vertices of value a and $-a$.*

Proof. Let $a = \max_{i \in V} |v_i|$. Suppose there exists a vertex i, such that $|v_i| < a$. Then, since G is connected, there must exist an edge between two vertices j and k, such that $|v_j| = a$ but $|v_k| < a$. This would mean

$$a = |v_j| = |(Mv)_j| = \left| \frac{1}{d} \sum_{\ell \in \Gamma(j)} v_\ell \right| \leq \frac{1}{d} \sum_{\ell \in \Gamma(j)} |v_\ell| = \frac{1}{d} \Big(|v_k| + \sum_{\ell \in \Gamma(j) \setminus \{k\}} |v_\ell| \Big) < \frac{1}{d} \sum_{\ell \in \Gamma(j)} a = a,$$

which is a contradiction.

Supose now there exists an edge between vertices j and k, where $v_j = v_k = a$. This is similarly a contradiction:

$$a = v_j = -(Mv)_j = -\frac{1}{d} \sum_{\ell \in \Gamma(j)} v_\ell = \frac{1}{d} \Big(v_k + \sum_{\ell \in \Gamma(j) \setminus \{k\}} v_\ell \Big) < -\frac{1}{d} \sum_{\ell \in \Gamma(j)} -a = a.$$

A similar contradiction can be shown when $v_j = v_k = -a$. □

Since $\lambda_{n-1} = -1$, there exist two non-parallel vectors that are (-1)-eigenvectors of M, say v and v'. We may assume that all coordinates of v and v' are ± 1. Let

$$S = \{i \in V | v_i = 1\}$$
$$S' = \{i \in V | v'_i = 1\}.$$

Since v and v' are non-parallel, $S' \neq S$ and $S' \neq V \setminus S$. We know that all edges cross between S and $V \setminus S$, as well as between S' and $V \setminus S'$. Therefore, $S \nabla S'$ and $V \setminus (S \nabla S')$ are disconnected, non-empty sides of the graph, which is a contradiction.

4b *(5 pts; do not start this problem until you are done with all others)* Prove that if $\lambda_n > -1/2$, then the graph G is not tripartite (recall that a graph is tripartite if its vertex set can be partitioned into three disjoint sets such that all of its edges connect vertices in different components).

Solution: We will prove the contrapositive: If G is bipartite, then $\lambda_n \leq -1/2$. We will instead look at the normalized Laplacian of G, $L = I - M$. We must prove that largest eigenvalue of L is at least 1.5 or alternately

$$\max_{0 \neq x \in \mathbb{R}^V} \frac{x^\top Lx}{x^\top x} \geq 1.5.$$

Recall that the quadratic form of the Laplacian is

$$x^\top Lx = \sum_{(u,v)\in E} (x_u - x_v)^2.$$

Since G is tripartite, we can partition V into $A \cup B \cup C$ such that there are no edges in $A \times A$, $B \times B$, or $C \times C$. Let the number of edges in $B \times C$, $C \times A$, and $A \times B$ be α, β, and γ respectively. Since G is d-regular we can note the relations

$$\begin{aligned}
&\alpha + \beta = dC, \\
&\beta + \gamma = dA, \\
&\gamma + \alpha = dB.
\end{aligned}$$

Consider the following three vectors in $\mathbb{R}^V$: w is $+1$ on A, -1 on B and 0 on C. y is $+1$ on B, -1 on C, and 0 on A. z is $+1$ on C, -1 on A, and 0 on B.

$$\begin{aligned}
&w^\top w = |A| + |B| \\
&y^\top y = |B| + |C| \\
&z^\top z = |C| + |A| \\
&w^\top Lw = \alpha + \beta + 4\gamma \\
&y^\top Ly = 4\alpha + \beta + \gamma \\
&z^\top Lz = \alpha + 4\beta + \gamma
\end{aligned}$$

From this we can deduce

$$\frac{w^\top Lw + y^\top Ly + z^\top Lz}{w^\top w + y^\top y + z^\top z} = 1.5,$$

so

$$\exists x \in \{w, y, z\} : \frac{x^\top Lx}{x^\top x} \geq 1.5,$$

as desired.

(This page is intentionally left blank.)

5 *(19 pts)* **Online ad allocation.** Alice and Bob started companies selling hand sanitizer, and are now advertising their products online to potential customers $c_1, c_2, \ldots, c_n$, where $c_i \in \mathcal{C}$ for all $i = 1, \ldots, n$. When a customer c_i arrives, they can be shown advertisement for either Alice's or Bob's hand sanitizer – we say that the customer is *allocated* to either Alice or Bob in that case. If S_1 and S_2 are the sets of customers allocated to Alice and Bob respectively at the end of the sequence, Alice will pay $v_1(S_1)$ Francs to the online advertisement engine and Bob will pay $v_2(S_2)$. Here $v_1 : 2^{\mathcal{C}} \to \mathbb{R}_+$ and $v_2 : 2^{\mathcal{C}} \to \mathbb{R}_+$ are non-negative monotone submodular functions. The goal in the online ad allocation problem is to design an allocation rule that maximizes $v_1(S_1) + v_2(S_2)$. In this problem you will analyze the competitive ratio of the greedy algorithm, stated below:

Algorithm 1 Greedy algorithm for online ad allocation

```
S_1 ← ∅, S_2 ← ∅
for i = 1, ..., n do
    if v_1(c_i|S_1) ≥ v_2(c_i|S_2) then
        S_1 ← S_1 ∪ {c_i}          ▷ Allocate i-th customer c_i to Alice
    else
        S_2 ← S_2 ∪ {c_i}          ▷ Allocate i-th customer c_i to Bob
    end if
end for
```

You will prove that greedy achieves a competitive ratio of $1/2$ by induction on n, the number of customers. We now describe the inductive step. Suppose that the first customer is allocated to Alice (the other case is analogous). Define for $S \subseteq \mathcal{C}$ the functions $v'_1(S) = v_1(S|\{c_1\})$ and $v'_2(S) = v_2(S)$, and let $p = v_1(\{c_1\})$. Let $\mathcal{I} = (v_1, v_2; c_1, \ldots, c_n)$ denote the input instance of the ad allocation problem, and let $\mathcal{I}' = (v'_1, v'_2; c_2, \ldots, c_n)$ denote the instance $\mathcal{I}$ with the first customer removed and the functions v_1, v_2 replaced with v'_1, v'_2. Let ALG denote the value achieved by greedy on $\mathcal{I}$, and let OPT denote the optimal offline solution on $\mathcal{I}$. Similarly, let ALG' denote the value achieved by greedy on $\mathcal{I}'$, and let OPT' denote the optimal offline solution on $\mathcal{I}'$.

5a *(5 pts)* Prove that $v'_j, j = 1, 2$, are non-negative monotone submodular functions.

5b *(10 pts)* Prove that $OPT \leq OPT' + 2p$.

5c *(4 pts)* Show how to complete the proof using **(b)**.

Solution:

5a Note that

$$\begin{aligned} v_1'(e \mid X) &= v_1'(\{e\} \cup X) - v_1'(X) \\ &= v_1(\{e\} \cup X \mid \{c_1\}) - v_1(X \mid \{c_1\}) \\ &= v_1(\{e, c_1\} \cup X) - v_1(\{c_1\}) - (v_1(\{c_1\} \cup X) - v_1(\{c_1\})) \\ &= v_1(\{e, c_1\} \cup X) - v_1(\{c_1\} \cup X) \\ &= v_1(e \mid \{c_1\} \cup X). \end{aligned}$$

Let X, Y be be such that $X \subseteq Y$. Note that for any element c, $X \cup \{c\} \subseteq Y \cup \{c\}$. Hence, we have

$$v_1'(e \mid X) = v_1(e \mid \{c_1\} \cup X) \geq v_1(e \mid \{c_1\} \cup Y) = v_1'(e \mid Y).$$

Thus v_1' satisfies diminishing returns property, and hence submodular. The non-negativity and monotonocity of v_1' follows from the monotonocity of v_1.

5b Let $O = (O_1, O_2)$ be the optimal allocation for I, so that $OPT = v_1(O_1) + v_2(O_2)$ and $OPT' = v_1'(O_1) + v_2'(O_2)$. Also note that $(O_1 \setminus \{c_1\}, O_2 \setminus \{c_1\})$ is a feasible solution on I' and moreover, since we assume c_1 is allocated to Alice, $v_1(\{c_1\}) \geq v_2(\{c_1\})$. Thus we have

$$\begin{aligned} OPT' &\geq v_1'(O_1 \setminus \{c_1\}) + v_2'(O_2 \setminus \{c_1\}) \\ &= v_1(O_1) - v_1(\{c_1\}) + v_2(O_2 \setminus \{c_1\}). \end{aligned}$$

Due to submodularity, we have $v_2(O_2 \setminus \{c_1\}) + v_2(\{c_1\}) \geq v_2(O_2) + v_2(\emptyset)$, and due to non-negativity, this yields $v_2(O_2 \setminus \{c_1\}) \geq v_2(O_2) - v_2(\{c_1\})$. Thus

$$OPT' \geq v_1(O_1) + v_2(O_2) - v_1(\{c_1\}) - v_2(\{c_1\}) \geq OPT - 2v_1(\{c_1\}) = OPT - 2p.$$

5c For $n = 1$, we get the optimal solution. Suppose $n > 1$. By the inductive hypothesis, $ALG' \geq OPT'/2$. Thus we have

$$ALG = p + ALG' \geq p + OPT'/2 = (2p + OPT')/2 \geq OPT/2.$$

6 *(14 pts)* **Learning from experts.** Recall that the hedge algorithm for learning from experts achieves the following guarantees in the setting of N experts with payoffs $\mathbf{m}^{(t)} \in [-1,+1]^N$ for $t = 1, 2, \ldots, T$. For every $\epsilon \in (0,1)$, if $\mathbf{p}^{(t)}$ for $t = 1, \ldots, T$ is the distribution picked by Hedge, then for every expert $i = 1, 2, \ldots, N$, one has $\sum_{t=1}^T \mathbf{p}^{(t)} \cdot \mathbf{m}^{(t)} \leq \sum_{t=1}^T m_i^{(t)} + \frac{\ln N}{\epsilon} + \epsilon T$. Minimizing the additive error term on the right hand side, we set $\epsilon = \sqrt{\frac{\ln N}{T}}$ and obtain

$$\sum_{t=1}^T \mathbf{p}^{(t)} \cdot \mathbf{m}^{(t)} \leq \sum_{t=1}^T m_i^{(t)} + 2\sqrt{T \ln N}.$$

In this problem you will prove that the above guarantee is essentially the best possible, showing that there exists a constant $c > 0$ such that no algorithm[2] can satisfy

$$\sum_{t=1}^T \mathbf{p}^{(t)} \cdot \mathbf{m}^{(t)} \leq \sum_{t=1}^T m_i^{(t)} + c\sqrt{T \ln N}. \tag{1}$$

For every $t = 1, 2, \ldots, T$ let $\mathbf{m}^{(t)} \in \{-1,+1\}^N$ denote a vector of independent Bernoulli random variables, and for every $i = 1, 2, \ldots, N$ let $X_i = \sum_{t=1}^T m_i^{(t)}$. Assume that N is bounded by a polynomial in T.

6a *(7 pts)* Prove that $\mathbb{E}[\min_{i=1,2,\ldots,N} X_i] = -\Omega(\sqrt{T \log N})$. You may use the following

Theorem 2 (Anti-concentration inequality for binomial random variables) *Let* $Y = \sum_{t=1}^T Y_t$*, where* Y_t *are independent random variables such that* $\Pr[Y_t = 1] = \Pr[Y_t = 0] = 1/2$*. Then for* $r \in [0, T/8]$

$$\Pr[Y < T/2 - r] \geq \frac{1}{15} e^{-16r^2/T}.$$

[2] Recall that an algorithm here is a method for choosing $\mathbf{p}^{(t)}$ as a function of $\mathbf{m}^{(s)}, s = 1, \ldots, t-1$, for every $t = 1, \ldots, T$.

6b *(7 pts)* Use **(a)** to prove that no algorithm can satisfy (1) for some absolute constant $c > 0$. *(Hint: recall that $\mathbf{p}^{(t)}$ is chosen before observing $\mathbf{m}^{(t)}$)*

Solution: **Part a.** We use Theorem 1 (anti-concentration bound) for X_i's. First, for each $i \in [N]$ and $t \in [T]$, let $n_i^{(t)} := \frac{m_i^{(t)}+1}{2}$, which implies that $n_i^{(t)} \in \{0, 1\}$. Now, let $Y_i := \sum_{t=1}^{T} n_i^{(t)}$. Now, for each $i \in [N]$, using Theorem 1, we have

$$\Pr\left[Y_i \leq \frac{T}{2} - \frac{1}{4}\sqrt{T \log N}\right] \geq \frac{1}{15} e^{-\frac{16T\log N}{16T}} = \frac{1}{15} e^{-\log N} = \frac{1}{15} N^{-1}.$$

Noting that $X_i = Y_i - \frac{T}{2}$, we get

$$\Pr\left[X_i \leq -\frac{1}{4}\sqrt{T \log N}\right] \geq \frac{1}{15} N^{-1}.$$

Consequently, if we define $Z := \min_{i=2,\ldots,N} X_i$, we have

$$\Pr\left[Z > -\frac{1}{4}\sqrt{T \log N}\right] \leq \left(1 - \frac{1}{15}N^{-1}\right)^{N-1} \leq e^{-\frac{1}{15}N^{-1}\cdot(N-1)} \leq e^{-\frac{1}{30}}.$$

Now, if we let $j = \mathrm{argmin}_{i\in[N]} X_i$, we have the following:

$$\begin{aligned}\mathbb{E}[X_j] = \mathbb{E}&\left[X_j | Z \leq -\frac{1}{4}\sqrt{T \log N}\right] \cdot \Pr\left[Z \leq -\frac{1}{4}\sqrt{T \log N}\right] \\ &+ \mathbb{E}\left[X_j | Z > -\frac{1}{4}\sqrt{T \log N}\right] \cdot \Pr\left[Z > -\frac{1}{4}\sqrt{T \log N}\right].\end{aligned}$$

At this point one should note that,

$$\begin{aligned}\mathbb{E}\left[X_j | Z > -\frac{1}{4}\sqrt{T \log N}\right] &= \mathbb{E}\left[\min\{X_1, Z\} | Z > -\frac{1}{4}\sqrt{T \log N}\right] \\ &\leq \mathbb{E}\left[X_1 | Z > -\frac{1}{4}\sqrt{T \log N}\right] \\ &= \mathbb{E}[X_1] = 0\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}[X_j] &\leq \mathbb{E}\left[X_j | Z \leq -\frac{1}{4}\sqrt{T \log N}\right] \cdot \Pr\left[Z \leq -\frac{1}{4}\sqrt{T \log N}\right] \\ &\leq -\frac{1}{4}\sqrt{T \log N} \cdot \left(1 - e^{\frac{1}{30}}\right) \\ &\leq -\frac{1}{8}\sqrt{T \log N}\end{aligned}$$

This implies that for constant $c = \frac{1}{8}$ we have

$$\mathbb{E}[X_j] \leq -c\sqrt{T \log N},$$

or equivalently (as stated in the question statement)

$$\mathbb{E}[X_j] = -\Omega\left(\sqrt{T \log N}\right).$$

Part b. Note that $\mathbf{p}^{(t)}$ is chosen before observing $\mathbf{m}^{(t)}$, as the hint suggests and by the way $m_i^{(t)}$'s are chosen, they are independent, so $\mathbb{E}\left[\sum_{t=1}^{T} \mathbf{p}^{(t)} \cdot \mathbf{m}^{(t)}\right] = 0$. Thus, again if we let $j = \text{argmin}_{i \in [N]} X_i$ and $c = \frac{1}{16}$, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbf{p}^{(t)} \cdot \mathbf{m}^{(t)} - \sum_{t=1}^{T} m_j^{(t)} - c\sqrt{T \log n}\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{p}^{(t)} \cdot \mathbf{m}^{(t)}\right] - \mathbb{E}\left[\sum_{t=1}^{T} m_j^{(t)}\right] - \frac{1}{16}\sqrt{T \log N} \\
&\geq 0 + \frac{1}{8}\sqrt{T \log N} - \frac{1}{16}\sqrt{T \log N} \\
&= \frac{1}{16}\sqrt{T \log N} > 0
\end{aligned}$$

where the first transition is due to **Part a**. This indicates that no matter what algorithm we use, there is a chance that we fail to satisfy the desired inequality.

(This page is intentionally left blank.)