Homework I, Algorithms II 2024

Solutions to many homework problems, including problems on this set, are available on the Internet, either in exactly the same formulation or with some minor perturbation. It is *not acceptable* to copy such solutions. It is hard to make strict rules on what information from the Internet you may use and hence whenever in doubt contact Michael Kapralov or Ola Svensson. You are, however, allowed to discuss problems in groups of up to three students.

Problem 1: Travel Itinerary (30 points)

A travel guide wants to create a travel itinerary. Suppose that there are n attractions, labeled $x_1, x_2, \ldots, x_n$, and m travelers, labeled $T_1, T_2, \ldots, T_m$. An *itinerary* consists of a list $L \subseteq \{x_1, x_2, \ldots, x_n\}$ of selected attractions. Each traveler T_j has a set of attractions that they *want to visit*, $V_j \subseteq \{x_1, x_2, \ldots, x_n\}$, and a set of attractions that they *do not want to visit*, $N_j \subseteq \{x_1, x_2, \ldots, x_n\}$. A traveler is considered *happy* if:

- At least one attraction they want to visit is in the itinerary (i.e., $\exists x_i \in V_j$ such that $x_i \in L$), **or**
- At least one attraction they don't want to visit is not included in the itinerary (i.e., $\exists x_i \in N_j$ such that $x_i \notin L$), so they feel that their preferences were considered.

A traveler will only join the trip if they are happy with the itinerary. Additionally, each traveler T_j has a discount coupon, so their payment for the trip is $w_j \geqslant 0$. The goal for the travel guide is to build an itinerary that maximizes revenue, i.e., to select an itinerary where the total payment from travelers who join the trip is as high as possible.

1a. *(10 points).* Argue that the following integer program finds an itinerary maximizing the revenue:

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{m} w_j z_j \\
\text{subject to} \quad & \sum_{i \in V_j} y_i + \sum_{i \in N_j} (1 - y_i) \geqslant z_j, \qquad \text{for all } j = 1, \ldots, m \\
& y_i \in \{0, 1\}, \qquad \text{for all } i = 1, \ldots, n, \\
& z_j \in \{0, 1\}, \qquad \text{for all } j = 1, \ldots, m.
\end{aligned}$$

1b. Let $(y_1^*, ..., y_n^*, z_1^*, ..., z_m^*)$ be an optimal solution to the LP relaxation of the integer program above, and consider a randomized rounding strategy that includes attraction x_i in the itinerary with probability y_i^*.

(i) *(10 points).* Show that the probability that traveler T_j is not happy can be upper bounded by $(1 - z_j^*/l_j)^{l_j}$ where $l_j = |V_j| + |N_j|$.

Hint. Use the Arithmetic-Geometric mean inequality (you don't need to prove it): For any nonnegative $a_1, \ldots, a_k$, it holds that

$$\left(\prod_{i=1}^{k} a_i\right)^{1/k} \leqslant \frac{1}{k}\sum_{i=1}^{k} a_i.$$

(ii) *(10 points).* Use this observation to show that the expected cost of the randomized rounding solution gives $(1 - 1/e)$-approximation to the value of maximal revenue.

You can use the following facts without proving them:

- $1 - (1 - x/k)^k \geqslant (1 - (1 - 1/k)^k)x$ for $x \in [0, 1]$, $k \in \mathbb{N}$.
- $1 - (1 - 1/k)^k \geqslant (1 - 1/e)$ for $k \in \mathbb{N}$.

Problem 2: Hedging for Max-Flow (30 points)

Let $G = (V, E)$ be an undirected graph with source $s \in V$ and sink $t \in V$. Let $\mathcal{P}$ denote the set of $s - t$ paths in G. Denote $|V| = n$. Consider the linear program formulation of the max-flow problem on this graph where each edge has capacity 1:

$$\begin{aligned} \text{maximize} \quad & \sum_{p \in \mathcal{P}} f_p \\ \text{subject to} \quad & \sum_{p \in \mathcal{P}: e \in p} f_p \leqslant 1 \quad \forall e \in E \\ & f_p \geqslant 0 \quad \forall p \in \mathcal{P}. \end{aligned}$$

This LP has exponentially many variables, so the standard methods for solving LPs can potentially run in exponential time. Therefore, your task is to implement the Hedge strategy to obtain an algorithm that for any $\epsilon \leqslant 1$ runs in time $poly(n)/\epsilon^2$ and returns $\{\bar{f}_p\}_{p \in \mathcal{P}}$ such that

- $\sum_{p \in \mathcal{P}} \bar{f}_p \geqslant OPT$, where OPT is the optimal solution of the max-flow LP and
- $\sum_{p: e \in p} \bar{f}_p \leqslant 1 + 2\epsilon$ for all $e \in E$. In other words, $\bar{f}$ almost satisfies the constraints of the LP.

Hint 1. You will need to design an efficient oracle for the reduced problem LP. Use a basic graph algorithm for that.

Hint 2. When obtaining bounds for T in the multiplicative weights framework, we need an upper bound ρ on the absolute value of $m_i^{(t)}$. It might happen that your $m_i^{(t)}$s are not bounded by $poly(n)$. However, observe that $OPT \leqslant n^2$, and use this fact to adapt your solution.

Problem 3: Perfect Matchings and Kittens (40 points)

In this problem, you will implement a polynomial time randomized[1] algorithm that:

- **takes as input** a bipartite graph $G = (A \cup B, E)$ with $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_n\}$ and $E \subseteq A \times B$, an integer weight function on the edges $w : E \to \{0, \ldots, w_{\max}\}$ with $w_{\max}$ being a positive integer (upper-bounded by some polynomial in n), and an integer $k \geqslant 0$;
- **outputs**
  - `yes` with probability at least $1 - 1/n$ if G has a perfect matching of weight exactly k;
  - `no` with probability 1 if G does not have a perfect matching of weight exactly k.

Let $\mathbb{F}$ be a finite field[2] such that $|\mathbb{F}| \geqslant \max\{w_{\max} \cdot n + 1, n^2\}$. Let $H(y, \mathbf{x})$ be an $n \times n$ symbolic matrix over variables $y, \mathbf{x}$ with $\mathbf{x} = (x_{i,j})_{i,j \in [n]}$ defined as

$$(H(y, \mathbf{x}))_{i,j} = \begin{cases} y^{w(a_i, b_j)} x_{i,j}\,, & \text{if } (a_i, b_j) \in E \\ 0\,, & \text{otherwise} \end{cases} \qquad \text{for all } i, j \in [n].$$

Consider the following algorithm.

1. Sample[3] $\boldsymbol{\alpha} \sim \text{unif}(\mathbb{F}^{n \times n})$.
2. Let $Y = \{\gamma_0, \ldots, \gamma_{n \cdot w_{\max}}\} \subseteq \mathbb{F}$ be such that $|Y| = n \cdot w_{\max} + 1$.
3. For each $j = 0, \ldots, n \cdot w_{\max}$ let $r_j = \det(H(\gamma_j, \boldsymbol{\alpha}))$, where $H(\gamma_j, \boldsymbol{\alpha}) \in \mathbb{F}^{n \times n}$ is the matrix obtained from $H(y, \mathbf{x})$ by substituting y with γ_j and $\mathbf{x}$ with $\boldsymbol{\alpha}$.
4. Interpolate the points $\{(\gamma_j, r_j)\}_{j=0}^{n \cdot w_{\max}}$ by writing a linear system $Pc = s$, where $P \in \mathbb{F}^{(n \cdot w_{\max}+1) \times (n \cdot w_{\max}+1)}$ and $s \in \mathbb{F}^{n \cdot w_{\max}+1}$ with $P_{j+1,i+1} = \gamma_j^i$ and $s_{j+1} = r_j$ for all $i, j \in \{0, \ldots, n \cdot w_{\max}\}$. Then, solve for $c \in \mathbb{F}^{n \cdot w_{\max}+1}$.
5. If $c_{k+1} \neq 0$ (where c_{k+1} is the degree-k coefficient) then output `yes`, otherwise output `no`.

3a. Theory (Optional, do not hand in) *(0 points).*

Analyze the correctness of the above algorithm.

3b. Implementation Assignment *(40 points).*

Solve the problem `ekjfhi2u3hfoiu` (`https://codeforces.com/group/3n0bzGpgms/contest/561054/problem/B`) on codeforces. Please see the PDF for instructions on how to submit your solution to the online judge and for grading criteria.

Hint. For improved efficiency, try shifting and rescaling the weights.

[1] Fun fact: it is not known whether there exists a polynomial time deterministic algorithm that distinguishes the `yes` and `no` case. A resolution to this open question would be a breakthrough result in Theoretical Computer Science, due to its implications on our understanding of the role of randomness in computation.

[2] For any prime number p, one has that $\mathbb{Z}_p$ is a finite field with p elements. When reading this section, you can then think of $\mathbb{F}$ as the integers modulo p for some p.

[3] This notation means that for each $i, j \in [n]$ we sample $\alpha_{i,j}$ independently and uniformly at random from $\mathbb{F}$.