Fine-grained Parameterized Algorithms

FPA · SoSe-2024 · tcs.uni-frankfurt.de/parameterized/ · 2024-07-03 · 2d88037

Problem set 6: Algebraic Methods

Overview: With this problem set, you can train reasoning about algebraic methods.

Instructions: For each skill, select **exactly one** problem below and submit your solution in Moodle; in your submission, make sure to repeat the problem that you are solving. The problems are roughly ordered by difficulty, choose a problem that you find non-trivial and interesting. (You are of course welcome to try the other problems as well and ask us for feedback.)

!! **Skill-6a. Reason about and adapt algebraic methods:** *I can formally reason about and adapt fast Möbius transforms and fast product operations.* (See Sections 10.1–10.3 in Cygan et al.)

easy **6.1 Proof of Proposition 10.10.** Prove that $\zeta = \sigma\mu\sigma$ and $\mu = \sigma\zeta\sigma$ hold.

6.2 Fast Packing Product. The *packing product* of two functions $f, g\colon 2^V \to \mathbb{Z}$ is a function $(f *_p g)\colon 2^V \to \mathbb{Z}$ such that for every $Y \subseteq V$, we have

$$(f *_p g)(Y) = \sum_{\substack{A,B\subseteq Y\\ A\cap B=\emptyset}} f(A)g(B)\,.$$

Show that all 2^n values of $f *_p g$ can be computed in time $2^n n^{O(1)}$, where $n = |V|$.

6.3 Möbius inversion on posets. In this guided exercise, we generalize the principle of Möbius inversion to finite partially ordered sets (posets). To this end, let P be a poset. The *incidence algebra* of a poset $(P, \leq)$ is defined as follows:

$$\mathbb{I}(P,\leq) := \{A \in \mathbb{C}^{P\times P} \mid x \not\leq y \Rightarrow A(x,y) = 0\}\,.$$

One example of an element of $\mathbb{I}(P,\leq)$ is the so-called *zeta function*:

$$\zeta(x,y) = \begin{cases} 1 & \text{if } x \leq y\,,\\ 0 & \text{otherwise.}\end{cases}$$

Consider the element μ of $\mathbb{I}(P,\leq)$, which is called the *Möbius function* over $(P,\leq)$ and which is inductively defined as follows:

$$\mu(x,y) = \begin{cases} 1 & \text{if } x = y\,,\\ 0 & \text{if } x \not\leq y\,,\\ -\sum_{x\leq z<y}\mu(x,z) & \text{otherwise.}\end{cases}$$

- Prove that the following identity holds for all $x, y \in P$:

$$\sum_{x\leq z\leq y} \mu(x,z) = \begin{cases} 1 & \text{if } x = y\\ 0 & \text{otherwise}\end{cases}$$

- Prove $\mu = \zeta^{-1}$ and conclude from $\zeta \cdot \mu = \mathrm{id}$ that the following identity holds as well:

$$\sum_{x \le z \le y} \mu(z, y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

- **Möbius inversion:** Let $f, g\colon P \to \mathbb{C}$ such that $g(x) = \sum_{y \le x} f(y)$ holds for all $x \in P$. Prove that the following identity holds for all $x \in P$:

$$f(x) = \sum_{y \le x} \mu(y, x) \cdot g(y)$$

!! <u>Skill-6b.</u> **Apply algebraic methods to design algorithms:** *I can apply inclusion–exclusion, the fast Möbius transform, and fast product operations to design fast algorithms.* (See Sections 10.1–10.3 in Cygan et al.)

6.4 Ryser's formula. Use the principle of inclusion–exclusion to design an algorithm which computes the number of perfect matchings in a given n-vertex bipartite graph in time $2^{n/2} n^{O(1)}$ and polynomial space.

6.5 List coloring. In the List Coloring problem, we are given an n-vertex graph G and, for each vertex $v \in V(G)$, there is a set (also called a list) of admissible colors $L(v) \subseteq \{1, \ldots, n\}$. The goal is to verify whether it is possible to find a proper vertex coloring $c\colon V(G) \to \mathbb{N}$ of G such that for every vertex v, we have $c(v) \in L(v)$. In other words, $L(v)$ is the set of colors allowed for v. Show a $2^n n^{O(1)}$-time algorithm for List Coloring. *Hint: Adapt the algorithm from Theorem 10.16.*

6.6 Counting subgraphs. In this guided exercise, you will develop an efficient algorithm for computing the number of subgraphs of a given graph G that are isomorphic to a fixed graph H.

Building on 6.3, we use Möbius inversion on the partition lattice: Let H be a graph with vertex set V. Given two partitions σ and ρ of V, we write $\sigma \to \rho$ if ρ can be obtained from σ by joining two blocks of σ. Example: For $\sigma = \{\{1, 4\}, \{2\}, \{3\}\}$, we have $\sigma \to \{\{1, 2, 4\}, \{3\}\}$. Now let $\le$ be the reflexive-transitive closure of $\to$, i.e., $\sigma \le \rho$ if and only if there are $\sigma_1, \ldots, \sigma_k$ with $\sigma \to \sigma_1 \to \cdots \to \sigma_k \to \rho$. Note that k might be zero.

- Let $P(V)$ be the set of partitions of V. Show that $(P(V), \le)$ is a poset. This poset is called the *partition lattice*. What is the minimum $\bot$ and the maximum $\top$ of this poset?
- Given an element $\sigma \in P(V)$, the graph H/σ is obtained from H by contracting each block of σ to a single vertex, deleting multiple edges, and keeping self-loops. Given a graph G, we let $\mathsf{Hom}(H, G)$ be the number of graph homomorphisms from H to G and let $\mathsf{Inj}(H, G)$ be the number of injective graph homomorphisms from H to G. Use Möbius inversion to prove

$$\mathsf{Inj}(H, G) = \sum_{\sigma \in P(V)} \mu(\bot, \sigma) \cdot \mathsf{Hom}(H/\sigma, G)\,. \tag{1}$$

- Given graphs H and G, it is known that the number of subgraphs of G that are isomorphic to H equals $\mathsf{Aut}^{-1}(H) \cdot \mathsf{Inj}(H, G)$, where $\mathsf{Aut}(H)$ is the number of automorphisms of H, that is, bijective homomorphisms from H to H. Combine this knowledge with Exercise 4.7 and (1) to design an algorithm for Subgraph Isomorphism. What is the running time of your algorithm?