Exact Algorithms

Homework 5

Lecturer: Eunjung KIM

1. Recall that a walk $W = v_1, \ldots, v_{k+1}$ is a sequence of vertices, not necessarily distinct, such that every $(v_i, v_{i+1}$ is an edge of G for $i = 1, \ldots, k$. Note that W is a walk from v_1 to v_{k+1}, and has length k. Let A be an adjacency matrix of a simple (no loop, no parallel edges) graph G over the vertex set $[n]$, i.e., the (i, j)-entry A_{ij}=1 if and only if (i, j) is an edge of G. For a positive integer k, let $M = A^k$. Show that for every $i, j = 1, \ldots, n$, the number of length k walks from i to j is equal to M_{ij}.

1-(a). Fill in the blank in the next statement "W is a walk of length $\ell + 1$ from i to j if and only if ( ) is a walk of length ℓ from i to k for some $k \in$ ( )."

1-(b). Let $t(i, j, \ell)$ be the number of length ℓ walks from i to j. Using the full sentence in 1-(a), complete the following recursion for t:

$$t(i, j, \ell + 1) = \sum_{k \in [n]} t(\quad, k, \quad) \cdot [(k, j) \in E(G)].$$

Here $[\cdot]$ takes value 1 if the sentence inside the bracket is true, 0 otherwise.

1-(c). Using the recursion in 2-(b), finalize the proof.

2. In the LIST k-COLORING problem, we are given a graph G and for each vertex $v \in V(G)$, there is a set (also called a *list*) of admissible colors $L(v)$. The goal is to verify whether it is possible to find a proper k-coloring $c : V(G) \to [k]$ 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 $2^n \cdot n^{O(1)}$-time algorithm for LIST k-COLORING.

Hint: define a k-tuple $(I_1, \ldots, I_k)$ of independent sets to be *L-eligible* if for $i \in [k]$, the vertex $v \in I_i$ has color i in its list $L(v)$. That is, a k-tuple of independent set is L-eligible if coloring each vertex set I_i by color i respects the admissible color list. The rest of the algorithm follows similar steps as in the algorithm for k-COLORING from the class. (The final dynamic programming algorithm needs a slightly different approach)

3. Let $G = (R \uplus C, E)$ be a bipartite graph with bipartition (R, C), $R = \{r_1, \ldots, r_n\}$ and $C = \{c_1, \ldots, c_n\}$. A matching M of G is said to *saturate* C if every vertex v of C is an endpoint of an edge in M. A *perfect matching* of G is a matching that saturates both R and C. Prove Ryser's formula which states that the number of perfect matchings in G equals

$$\sum_{X \subseteq R} (-1)^{n-|X|} \prod_{j \in C} \sum_{i \in X} a_{ij}$$

where a_{ij} is an (i, j)-entry of the bi-adjacency[1] matrix A of G.

[1] The rows of A are identified with R and the columns of A are identified with C. And the entries of A are defined as: $a_{ij} = 1$ if $i \in R$ and $j \in C$ are adjacent in G, and $a_{ij} = 0$ otherwise.

Hint: Use Inclusion-Exclusion principle. Define the universe $\mathcal{U}$ as the set of all n-tuples of edges $(e_1, \ldots, e_n)$ such that the endpoint of e_i in C is c_i.

♣ Submit your solution via email (eunjungkim78@gmail.com) by 20 Feb 2019, midnight.
♣ Questions with ★ can be worked together with a colleague. But please write the solution by yourself.