Exact Algorithms

Homework 2

Lecturer: Eunjung KIM

1. Given a graph G and an integer k, the problem EDGE BIPARTIZATION asks to find an edge subset $D \subseteq E(G)$, called *edge bipartization set*, of size at most k so that $G - D$ becomes a bipartite graph. The goal of this exercise is to design an iterative compression algorithm for EDGE BIPARTIZATION with runtime $2^k \cdot n^{O(1)}$. An *odd cycle transversal* is a vertex set Z of G such that $G - Z$ is bipartite. For each of (a)-(c), present a runtime analysis and a proof of correctness of the algorithm as well. You would want to use an algorithm for finding min-(s,t)-cut as a subroutine, for which you can assume a black-box polynomial-time algorithm is given.

1(a). Design a $4^k \cdot n^{O(1)}$-time algorithm for the following COMPRESSION EDGE BIPARTIZATION problem: given a graph $G = (V, E)$ and an edge bipartization set X of size at most $k + 1$, find an edge bipartization set X' with strictly smaller size, or correctly decide that there is no such solution.

1(b). Design a $2^k \cdot n^{O(1)}$-time algorithm for the problem EDGE BIPARTIZATION/OCT: given a graph $G = (V, E)$ and an odd cycle transversal X of size at most $k + 1$, find an edge bipartization set X of size at most k, or correctly decide that there is no such solution.

1(c) Present an iterative compression algorithm for the EDGE BIPARTIZATION of running time $2^k \cdot n^{O(1)}$.

2. In the PARTIAL VERTEX COVER problem, we are given an undirected graph G and positive integers k and t, and the goal is to check whether there exists a set $X \subseteq V(G)$ of size at most k such that at least t edges of G are incident to vertices on X. Obtain an algorithm with running time $2^{O(t)} \cdot n^{O(1)}$ for the problem. You may want to use Stirling's approximation; $n! \approx (\frac{n}{e})^n$.

3. Consider the following problem (k, q)-SEPARATOR: given an undirected graph G and positive integers k and q, find a set at most k vertices such that $G - X$ has at least two components of size at least q.

3(a). Consider a random coloring of the vertices of G with two colors L and R. Evaluate the probability that all X vertices belong to L and q vertices of each of two "large" components belong to R.

3(b). Suppose that L and R is a partition of $V(G)$ as depicted in 2(a). Prove that there exists two vertices $s, t \in R$ and a set of vertices $Y \subseteq L$ of size at most k such that s and t belongs to distinct component in $G - Y$.

3(c). A minimum (s,t)-cut of an edge-weighted directed graph D (possibly with 2-cycles) is a set S of edges such that $D - S$ does not contain a directed path from s to t. Using a polynomial-time algorithm $\mathcal{A}$ solving finding a min (s,t)-cut as a black box, present a polynomial time algorithm for the following problem: given an undirected graph G and two distinct, non-adjacent vertices s and t, find a minimum-size set of vertices $Y \subseteq V(G) \setminus \{s,t\}$ such that s and t belongs to distinct components in $G - Y$.

3(d). Present an algorithm for solving (k, q)-SEPARATOR in expected running time $2^{O(q+k)} \cdot n^{O(1)}$.

4. For an $n \times n$ matrix A with (i, j)-th entry a_{ij}, the *permanent* of A is defined as $\sum_{i=1}^{n} \prod_{\sigma} a_{i\sigma(i)}$ where σ runs over all permutations on $[n]$. Design a dynamic programming algorithm for computing the permanent of A in time $2^n \cdot n^{O(1)}$.

5. 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 using the inclusion-exclusion based approach.

6. 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.

Hint: Use Inclusion-Exclusion formula. 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 4 Feb 2021, midnight.

[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.