PS Complexity of Polynomial-Time Problems

https://www.cosy.sbg.ac.at/~sk/courses/polycomp/

Exercise sheet 3 — Due: Sunday, December 3, 2017

Total points : 40

Prove all your claims!

Exercise 1 (*10 points*)

The **Metricity** problem is defined as follows: Given an $n \times n$ matrix A with entries in $\{0, \dots, \lfloor n^c \rfloor\}$ for some constant $c > 0$, decide whether $A[i, j] \leq A[i, k] + A[k, j]$ for all $i, j, k \in \{1, \dots, n\}$. Prove that **Metricity** is equivalent to **APSP** under subcubic reductions.

Hint: *Reduce* ***Metricity*** *to* ***Min-Plus Product*** *and reduce* ***Negative Triangle*** *to* ***Metricity****.*

Exercise 2 (*10 points*)

Consider a directed graph $G = (V, E, w)$ with positive integer weights in the range $w(e) \in \{1, 2, \dots, W\}$ for each edge $e \in E$. The **Betweenness Centrality** of a node $v \in V$ is the number of pairs s, t such that v lies on a shortest path from s to t, i.e.,

$$\mathrm{BC}_G(v) = \Big|\{(s,t) \,:\, s, t \in V \setminus \{v\}, s \neq t, \mathrm{dist}(s,t) = \mathrm{dist}_G(s,v) + \mathrm{dist}_G(v,t)\}\Big| .$$

and the **Diameter** of G is the maximum distance between any pair of nodes, i.e.,

$$\mathrm{diam}(G) = \max_{s,t \in V} \mathrm{dist}_G(s,t) .$$

Show that if there is an algorithm for computing the **Betweenness Centrality** of a node in a graph with positive edge weights running in time $T(n, m)$, then there is an algorithm for computing the diameter of a graph with positive edge weights running in time

$$O\Big(T\big(O(n), O(m+n)\big) \log(nW) + m\Big) .$$

Hint: *Introduce a* dummy node *and perform binary search to find the value of the diameter.*

Exercise 3 (*10 points*)

Give an algorithm for **Orthogonal Vectors** with running time $O(n^\omega)$, i.e., an algorithm that (theoretically) outperforms the naive $O(n^2 d)$-time algorithm in the high-dimensional regime.

Exercise 4 (*10 points*)

Work out the subcubic reduction from **All-Pairs Triangle Detection** to **Triangle Detection** mentioned in class. Note: As a consequence, this will prove that if there is a subquadratic *combinatorial* algorithm for **Triangle Detection**, then there also is a subquadratic *combinatorial* algorithm for **All-Pairs Triangle Detection**.