HW 4 (due Monday, 6pm, October 5, 2015)

NEW CS 473: Theory II, Fall 2015 Version: **1.0**

Collaboration Policy: For this homework, Problems 1–2 can be worked in groups of up to three students. Submission is online on moodle.

---

1. (30 PTS.) TSP for k-silly graphs.

You are given the a graph G over the set of vertex $\mathsf{V} = [\![n]\!]$, with the edge ij having weight $w(i,j) > 0$, for all $i < j$.

Given a parameter $k > 0$, describe an algorithm, as fast as possible, that computes the exact shortest TSP in G, assuming that the TSP can use an edge ij only if $|i-j| \leq k$. (For a fixed k the running time of your algorithm should be polynomial.)

[The recursive subproblem here is somewhat messy, but should be doable after the last homework. Figure it out, and the rest should be easy.]

2. (70 PTS.) Packing things.

(A) (10 PTS.) Let $\mathcal{I}$ be a given set of n closed intervals on the real line, and let $k > 0$ be a parameter. A ***k-packing*** of $\mathcal{I}$ is a set of intervals $\mathcal{J} \subseteq \mathcal{I}$, such that no point is contained in more than k intervals of $\mathcal{J}$.

Describe an algorithm, as efficient as possible, that computes the largest subset of $\mathcal{I}$ that is a k-packing. (For full credit your algorithm has to run in polynomial time in k and n.)

[Hint: Use a greedy algorithm and **prove** that it indeed outputs the optimal solution in this case. If you are unable to do the proof (which is a bit subtle) – no worries, you can still use the algorithm as a black box in the later parts of this problem.]

(B) (30 PTS.) Let $\mathcal{R}$ be a given set of axis-parallel rectangles in the plane, where the ith rectangle is of the form $[a_i, b_i] \times [c_i, d_i]$.

A ***k-packing*** is a set of rectangles $\mathcal{Q} \subseteq \mathcal{R}$, such that no point is contained in more than k rectangles of $\mathcal{Q}$.

Describe an approximation algorithm, as efficient as possible, that outputs a k-packing of $\mathcal{R}$ of size $\geq \mathrm{OPT}/t$, where t is as small as possible and OPT is the size of the largest k-packing of $\mathcal{R}$. What is the value of t of your algorithm in the worst case? What is the running time of your algorithm?

Provide a self contained proof of the approximation quality of your algorithm.

[Hint: See lecture slides.]

(C) (30 PTS.) Let $\mathcal{B}$ be a given set of axis-parallel boxes in three dimensions, where the ith box is of the form $[a_i, b_i] \times [c_i, d_i] \times [e_i, f_i]$.

A ***k-packing*** is a set of boxes $\mathcal{C} \subseteq \mathcal{B}$, such that no point is contained in more than k boxes of $\mathcal{C}$.

Describe an approximation algorithm, as efficient as possible, that outputs a k-packing of $\mathcal{B}$ of size $\geq \mathrm{OPT}/t$, where t is as small as possible and OPT is the size of the maximum k-packing of $\mathcal{B}$. What is the value of t of your algorithm in the worst case? What is the running time of your algorithm?

Provide a self contained proof of the approximation quality of your algorithm.

[Hint: Use (B).]