INTRO TO APPROXIMATION – HW3

Deadline: Sunday **February 4, 2018 23:59 AoE**. Send your solutions preferably by email (in PDF, ODT ... or just inside the email). I also accept scans or photos of quality high-enough to read everything without problems. Please ask if any homework is not clear to you or if you think that the description below is missing something.

Homework one — Parallel greedy graph coloring [5 points]

Consider graphs with the maximum degree at most Δ. A greedy algorithm can color any such graph easily with $\Delta + 1$ colors – but it is not easy to see how to parallelize it. Suggest a fast randomized parallel algorithm, which can properly color such a graph with $\Delta+1$ colors with very high probability.

Think of Δ as a parameter of your algorithm – your parallel computers know Δ in advance. However, Δ can be as large as n, the number of vertices.

Tip: You may want to build on the material from the lectures.

Homework two — A simpler almost 2-universal hash functions [5 points]

Let n be an integer and $p \geq n$ be a prime. In the class we showed that the family

$$\mathcal{H} = \{h_{a,b} | 1 \leq a \leq p-1, 0 \leq b \leq p-1\}$$

is (weakly) 2-universal where $h_{a,b} = (ax + b \mod p) \mod n$.

Consider now hash functions $h_a = (ax \mod p) \mod n$ and family

$$\mathcal{H}' = \{h_a | 1 \leq a \leq p-1\}.$$

Give an example to show that $\mathcal{H}'$ is not 2-universal and then prove that $\mathcal{H}'$ is almost 2-universal in the following sense: for any $x, y \in \{0, 1, \ldots, p-1\}, x \neq y$, if h is chosen uniformly at random from $\mathcal{H}'$ then

$$\Pr[h(x) = h(y)] \leq \frac{2}{n}.$$

Homework three — Lower bound for strongly 2-universal hash functions using LA [6 points]

From the lecture we might recall what strongly 2-universal hash functions are: A family $\mathcal{H}$ of functions (all being hash functions, i.e., $h\colon U \to HT$) is strongly 2-universal, if for any quadruple of values $x, x' \neq x, y, y'$ it is true that

$$\Pr_{h \text{ randomly chosen from } \mathcal{H}}[h(x) = y \wedge h(x') = y'] = \frac{1}{|HT|^2}.$$

You might know that for every $|U| = 2^n$, $|HT| = 2^m$ we can generate a (weakly or strongly) 2-universal hash function using only $O(m+n)$ fully random bits. This is asymptotically tight, and the following sequence of observations proves this. We will use linear algebra to show this.

We now assume that $\mathcal{H}$ is any strongly 2-universal hash family.

1. Prove that if $|U| \geq 2$, then $|\mathcal{H}| \geq |HT|^2$.
2. Prove that if $|HT| = 2$, then $|\mathcal{H}| \geq |U| + 1$.

   Tip: Using functions from $\mathcal{H}$, generate for each x one vector v_x from $\mathbb{R}^{|\mathcal{H}|}$ with coordinates of type ± 1, while making sure that every pair v_x and $v_{x'}$ is orthogonal. If we find such vectors, what does it mean for the dimension of $\mathbb{R}^{|\mathcal{H}|}$? And why is the right side $|U|+1$ and not just $|U|$?

3. Generalize the previous approach and prove that for arbitrary $|HT|$ it holds that

$$|\mathcal{H}| \geq |U|(|HT|-1)+1.$$

Tip: Using functions from $\mathcal{H}$, generate for each x a set of $|HT|-1$ different vectors $v_{x,y} \in \mathbb{R}^{\mathcal{H}}$ so that for one x all vectors $v_{x,y}$ are linearly independent, and for two different $x \neq x'$ and any y, y' we have that the vectors $v_{x,y}$ and $v_{x',y'}$ are even orthogonal. What does this mean for the dimension of $\mathbb{R}^{|\mathcal{H}|}$?

4. Use the previous bullet points and conclude that if $|U| = 2^n$ and $|HT| = 2^m$, then for generating a random function from $\mathcal{H}$ we need at least $(\max(n,m)+m)$ random bits, which is asymptotically tight.

HOMEWORK FOUR — k-SUPPLIER PROBLEM [5 points]

In the metric k-SUPPLIER PROBLEM we get $m+n$ points on input, where m of them are (in advance) marked as *suppliers* and the rest are *consumers*. Between all those points is a metric (with a triangle inequality as usual). Our task in this case is to select k suppliers so that we minimize the longest distance between a customer and its closest supplier.

Suggest and analyze a 3-approximation algorithm for the k-SUPPLIER PROBLEM.

Tip: This problem is related to the k-CENTER PROBLEM about which you can read in the Williamson, Shmoys book, chapter 2.2. (available online, see a link on the webpage).

HOMEWORK FIVE — Bonus exercise: implementation of a TSP approximation algorithm [7 points]

Solve the first or the second exercise from the 3rd HW set last year: `http://iuuk.mff.cuni.cz/~bohm/16-17/apxintro/03-hw-en.pdf`. That is, write an implementation of the 2-approximation algorithm for TSP which uses doubling of the minimum spanning tree, or write an implementation of the Christofides algorithm. The former is worth 4 points, while the latter is worth 7 points.

The exactly same conditions as last year apply (except that the deadline is as above and the number of points is higher). In particular, collaboration on the solution is not allowed this time.