ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Final Exam, Advanced Algorithms 2016-2017

- You are only allowed to have a handwritten A4 page written on both sides.
- Communication, calculators, cell phones, computers, etc... are not allowed.
- Your explanations should be clear enough and in sufficient detail that a fellow student can understand them. In particular, do not only give pseudo-code without explanations. A good guideline is that a description of an algorithm should be such that a fellow student can easily implement the algorithm following the description.
- **You are allowed to refer to material covered in the course** including theorems without reproving them.
- **Do not touch until the start of the exam.**

Good luck!

Name: ______________________ **N° Sciper:** ______________

Problem 1 / 20 points	Problem 2 / 20 points	Problem 3 / 20 points	Problem 4 / 20 points	Problem 5 / 20 points

Total / 100

1 *(consisting of subproblems **a**-**b**, 20 pts)* **Basic questions.** This problem consists of two subproblems that are each worth 10 points.

1a *(10 pts)* **LSH for Jaccard similarity.**

Recall the Jaccard index that we saw in Exercise Set 10: Suppose we have a universe U. For non-empty sets $A, B \subseteq U$, the Jaccard index is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}\,.$$

Design a locality sensitive hash (LSH) family $\mathcal{H}$ of functions $h : 2^U \to [0, 1]$ such that for any non-empty sets $A, B \subseteq U$,

$$\Pr_{h \sim \mathcal{H}}[h(A) \neq h(B)] \begin{cases} \leq 0.01 & \text{if } J(A, B) \geq 0.99, \\ \geq 0.1 & \text{if } J(A, B) \leq 0.9. \end{cases}$$

(In this problem you are asked to explain the hash family and argue that it satisfies the above properties. Recall that you are allowed to refer to material covered in the course.)

Solution:

1b *(10 pts)* **Randomized algorithms for min-cut.** In class, we saw Karger's beautiful randomized algorithm for finding a min-cut in an undirected graph $G = (V, E)$ with $n = |V|$ vertices. Each iteration of Karger's algorithm can be implemented in time $O(n^2)$, and if repeated $\Theta(n^2 \log n)$ times, Karger's algorithm returns a min-cut with probability at least $1 - 1/n$. However, this leads to the often prohibitively large running time of $O(n^4 \log n)$.

Karger and Stein made a crucial observation that allowed them to obtain a much faster algorithm for min-cut: the Karger-Stein algorithm runs in time $O(n^2 \log^3 n)$ and finds a min-cut with probability at least $1 - 1/n$.

Explain in a couple of sentences the main idea that allowed Karger and Stein to modify Karger's algorithm into the much faster Karger-Stein algorithm. In other words, what are the main differences between the two algorithms?

Solution:

(This page is intentionally left blank.)

2 *(20 pts)* **Set packing in the semi-streaming model.**

Consider the problem of finding a maximum cardinality set packing in the semi-streaming model. An instance of this problem consists of a known universe U of n elements and sets $S \subseteq U$ are streamed one-by-one. The goal is to select a family $\mathcal{T}$ of pairwise disjoint sets (i.e., $S \cap S' = \emptyset$ for any two distinct sets $S, S' \in \mathcal{T}$) of maximum cardinality while only using $O(n \cdot \text{poly}\log n)$ storage space.

Devise an algorithm in this setting that returns a set packing of cardinality at least $1/k$ times that of a maximum cardinality set packing, assuming that each streamed set S has cardinality at most k, i.e., $|S| \leq k$.

(In this problem you are asked to (i) design the algorithm, (ii) show that it uses $O(n \cdot \text{polylog}\, n)$ space, and (iii) prove that it returns a solution of cardinality at least $1/k$ times the cardinality of a maximum cardinality set packing. Recall that you are allowed to refer to material covered in the course.)

Solution:

(This page is intentionally left blank.)

3 *(20 pts)* **Amplifying success probability of an unbiased estimator.**

Professor Ueli von Gruyères has worked intensely throughout his career to get a good estimator of the yearly consumption of cheese in Switzerland. Recently, he had a true breakthrough. He was able to design an incredibly efficient randomized algorithm $\mathcal{A}$ that outputs a random value X satisfying

$$\mathbb{E}[X] = c \qquad \text{and} \qquad \mathrm{Var}[X] = c^2 \,,$$

where c is the (unknown) yearly consumption of cheese in Switzerland. In other words, $\mathcal{A}$ is an unbiased estimator of c with variance c^2.

Use Ueli von Gruyères' algorithm $\mathcal{A}$ to design an algorithm that outputs a random value Y with the following guarantee:

$$\Pr[|Y - c| \geq \epsilon c] \leq \delta \qquad \text{where } \epsilon > 0 \text{ and } \delta > 0 \text{ are small constants.} \tag{1}$$

Your algorithm should increase the resource requirements (its running time and space usage) by at most a factor $O(1/\epsilon^2 \cdot \log(1/\delta))$ compared to the requirements of $\mathcal{A}$.

(In this problem you are asked to (i) design the algorithm using $\mathcal{A}$, (ii) show that it satisfies the guarantee (1)*, and (iii) analyze how much the resource requirements increase compared to that of simply running $\mathcal{A}$. Recall that you are allowed to refer to material covered in the course.)*

Solution:

(This page is intentionally left blank.)

4 *(20 pts)* **Min-weight perfect matching via determinants.** Consider the following algorithm that takes as input a complete n-by-n bipartite graph $G = (U \cup V, E)$ with positive integer edge-weights $w : E \to \mathbb{Z}_{>0}$:

MinWeightPerfectMatching(G, w):

1. **for** each edge $e \in E$ (i.e., each pair (u, v) since the graph is complete)
2.   select independently and uniformly at random $p(e) \in \{1, \ldots, n^2\}$.
3. Define a bi-adjacency matrix A with n rows (one for each $u \in U$) and n columns (one for each $v \in V$) as follows:

$$A_{u,v} = 2^{n^{100} w(u,v)} \cdot p(u, v)\,.$$

4. **return** largest positive integer i such that $2^{i \cdot n^{100}}$ divides $\det(A)$ (if no such i exists, we return 0).

Prove that the above algorithm returns the value of a min-weight perfect matching with probability at least $1 - 1/n$. Recall that you are allowed to refer to material covered in the course.

Hint: Let $\mathcal{M}_i$ denote the set of perfect matchings M whose weight $\sum_{e \in M} w(e)$ equals i. Use that one can write $\det(A)$ as follows:

$$\det(A) = \sum_{i=0}^{\infty} 2^{i \cdot n^{100}} f_i(p) \qquad \text{where } f_i(p) = \sum_{M \in \mathcal{M}_i} \operatorname{sign}(M) \prod_{e \in M} p(e)\,.$$

Here $\operatorname{sign}(M) \in \{\pm 1\}$ is the sign of the permutation corresponding to M.

Solution:

(This page is intentionally left blank.)

5 *(20 pts)* **Assigning vertex potentials.** Design and analyze a polynomial time algorithm for the following problem:

INPUT: An undirected graph $G = (V, E)$.

OUTPUT: A non-negative vertex potential $p(v) \geq 0$ for each vertex $v \in V$ such that

$$\sum_{v \in S} p(v) \leq |E(S, \bar{S})| \quad \text{for every } \emptyset \neq S \subsetneq V \quad \text{and} \quad \textstyle\sum_{v \in V} p(v) \text{ is maximized.}$$

(Recall that $E(S, \bar{S})$ denotes the set of edges that cross the cut defined by S, i.e., $E(S, \bar{S}) = \{e \in E : |e \cap S| = |e \cap \bar{S}| = 1\}$.)

Hint: formulate the problem as a large linear program (LP) and then show that the LP can be solved in polynomial time.

(In this problem you are asked to (i) design the algorithm, (ii) show that it returns a correct solution and that it runs in polynomial time. Recall that you are allowed to refer to material covered in the course.)

This year we didn't cover the Elliposid method which is necessary for solving the above problem. By using the Ellipsoid method, the above reduces to the following: Given p, give an efficient algorithm that either verifies that p is feasible or outputs a violated constraint.

Solution:

(This page is intentionally left blank.)