# Final Exam, Advanced Algorithms 2021-2022

- 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 lecture notes** including theorems without reproving them.
- **Problems are not necessarily ordered by difficulty.**
- **Do not touch until the start of the exam.**

**Good luck!**

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

| Problem 1<br>/ 30 points | Problem 2<br>/ 20 points | Problem 3<br>/ 20 points | Problem 4<br>/ 15 points | Problem 5<br>/ 15 points |
|---|---|---|---|---|
| | | | | |

| **Total / 100** |
|---|
| |

**1** *(*30 pts) **Linear Programming and Duality.** In the 2-edge connected subgraph problem we are given an undirected graph $G = (V, E)$ with edge costs $c_e \geq 0, e \in E$. The goal is to find the cheapest 2-edge connected subgraph of $G$, i.e. a subset of edges $F \subseteq E$ such that every cut in $G_F = (V, F)$ has size at least two. We consider the linear programming relaxation of the 2-edge connected subgraph problem.

For a set $S \subseteq V$ we write $\delta(S)$ to denote the set of edges with exactly one endpoint in $S$. The linear program is

$$\begin{aligned} \min \sum_{e \in E} & c_e \cdot x_e \\ \text{s.t.} \sum_{e \in \delta(S)} & x_e \geq 2 \quad \text{for every } S \subset V, S \neq \emptyset \\ & x_e \geq 0 \text{ for all } e \in E \end{aligned}$$

**1a** *(15 pts)* Give a polynomial time algorithm that checks whether a given candidate solution $x = (x_e)_{e \in E}$ is feasible.

**1b** *(15 pts)* Write down the dual linear program to the linear program above.

**2** *(*20 pts) **Fair coins and how to find them.** Alice has $n$ coins, of which one is a fair coin (it falls heads with probability $1/2$ and tails with probability $1/2$) and the others are $\varepsilon$-biased (they all fall heads with probability at least $1/2+\varepsilon$). Prove that Alice can determine the fair coin with probability at least $9/10$ after flipping the coins $O(\frac{1}{\varepsilon^2} n \log n)$ times in total.

*Hint: first prove that you can estimate the bias of any fixed coin to additive precision $\varepsilon$ with sufficiently high certainty using $O(\frac{1}{\varepsilon^2} \log n)$ coin flips. Use a Chernoff bound for that.*

**Solution to problem 2 continued.**

**3** *(20 pts)* **Streaming Approximation to MAX-CUT.** In the MAX-CUT problem one is given a graph $G = (V, E)$ with $n$ vertices and $m$ edges, and the task is to find

$$\gamma_{MAX-CUT}(G) := \max_{S \subseteq V} E(S, V \setminus S),$$

where we let $E(S, V \setminus S)$ denote the number of edges that cross the cut $(S, V \setminus S)$ in $G$.

In this problem you will design a single-pass streaming algorithm that achieves a $(1 + \varepsilon)$ approximation to $\gamma_{MAX-CUT}(G)$ using $O(\frac{1}{\varepsilon^2} n \log n)$ bits of space. Specifically, your algorithm should output $\gamma_{ALG}(G)$ such that with probability at least $9/10$ over the algorithm's internal randomness one has

$$(1 - \varepsilon)\gamma_{MAX-CUT}(G) \leq \gamma_{ALG}(G) \leq (1 + \varepsilon)\gamma_{MAX-CUT}(G).$$

Your algorithm should take a single pass over an adversarially ordered stream of edges of $G$. You may assume knowledge of $n$ and $m$ (and, of course, knowledge of $\varepsilon$). You may assume that your algorithm has a source of random bits. The algorithm need not be polynomial time.

*Note: you can use any result from any of the two homework assignments without reproving it; also, it will be useful to show that* $\gamma_{MAX-CUT}(G) \geq m/2$ *for every graph* $G$.

**Solution to problem 3 continued.**

**4** (15 pts) **Distributed cut recovery.** For a graph $G = ([n], E), [n] = \{1, 2, \ldots, n\}$, let $B \in \{-1, 0, +1\}^{\binom{n}{2} \times n}$ denote the signed edge-vertex incidence matrix of $G$, defined as follows. The rows of $B$ are indexed by pairs of vertices of $G$ (i.e., by potential edges) and columns are indexed by vertices of $G$. For every $e = \{i, j\}, i, j \in \{1, 2, \ldots, n\}, i < j$, and every $k \in [n]$ one has

$$B_{e,k} = \begin{cases} 1 & \text{if } k = i \\ -1 & \text{if } k = j \\ 0 & \text{o.w.} \end{cases}$$

if $e \in E$ and $B_{e,k} = 0$ for all $k \in [n]$ otherwise.

**4a** (5 pts) For every $S \subseteq [n]$ let $\mathbf{1}_S \in \{0, 1\}^n$ denote the indicator vector of of $S$, i.e. a vector with 1's in positions corresponding to vertices in $S$ and 0's elsewhere. Prove that the nonzeros of the vector $B\mathbf{1}_S$ are exactly the edges in $G$ that cross the cut $(S, V \setminus S)$.

**4b** (10 pts) Now suppose that Alice holds the first $n/2$ columns of $B$, Bob holds the remaining $n/2$ columns of $B$, and Charlie holds $S \subset V$ such that the cut $(S, V \setminus S)$ contains at most $r$ edges. Show that Alice and Bob can each send Charlie a message of $O(nr \log^2 n)$ bits such that from the two messages Charlie can reconstruct the edges that cross the cut $(S, V \setminus S)$ in $G$, with probability $9/10$. You may assume that Alice, Bob and Charlie share a source of random bits.

*Hint: you can use CountSketch to solve this problem, but other solutions also exist.*

**5** *(15 pts)* **Maximum independent set.** A subset $S$ of vertices of a graph $G = (V, E), |V| = n, |E| = m$, is independent in $G$ if no edge in $E$ has both endpoints in $S$. The maximum independent set in $G$ is an independent set of largest size in $G$. Consider the following linear program:

$$\max \sum_{v \in V} x_v$$

s.t.

$$\begin{aligned} x_u + x_v &\le 1 \qquad && \forall e = \{u, v\} \in E \\ x_v &\ge 0 && \forall v \in V \end{aligned}$$

Note that this is a relaxation of the maximum independent set problem: for every independent $S$ we can let $x_v = 1$ for $v \in S$ and $x_v = 0$ for $v \in V \setminus S$, obtaining a feasible solution to the above LP whose value equals the size of $S$.

Let $x^*$ denote the optimal solution to the LP above. We round $x^*$ to an integral solution as follows. If the value of the LP is at most $2\sqrt{m}$, output the singleton set containing any vertex of $G$. Otherwise construct an independent set in $G$ as follows. First let $S$ contain every vertex $v \in V$ independently with probability $x_v^*/\sqrt{m}$. If for some edge $e = \{u, v\} \in E$ both $u$ and $v$ belong to $S$, remove both $u$ and $v$ from $S$. Denote the resulting set by $S_{ALG}$ (note that $S_{ALG}$ is an independent set in $G$).

In this problem you will prove that $\mathbb{E}[|S_{ALG}|] \ge \frac{1}{2\sqrt{m}} |S_{OPT}|$, where $S_{OPT}$ is the largest independent set in $G$.

**5a** *(5 pts)* Prove that $\mathbb{E}[|S_{ALG}|] \ge \sum_{v \in V} x_v^*/\sqrt{m} - \sum_{u,v \in V: \{u,v\} \in E} x_u^* x_v^*/m$.

**5b** *(5 pts)* Prove that $\sum_{u,v \in V : \{u,v\} \in E} x_u^* x_v^* / m \leq 1$.

**5c** *(5 pts)* Using the results of **5a** and **5b** conclude that $\mathbb{E}[|S_{ALG}|] \geq \frac{1}{2\sqrt{m}} |S_{OPT}|$.