Due: Thursday, April 16, 2020 4:59PM on MarkUs

You will receive 20% of the points for any (sub)problem for which you write "I do not know how to answer this question." You will receive 10% if you leave a question blank. If instead you submit irrelevant or erroneous answers you will receive 0 points. You may receive partial credit for the work that is clearly "on the right track." You may choose to spend your time looking for solutions on the internet and may likely succeed in doing so but you probably won't understand the concepts. So at the very least try to do the assignment initially without searching the internet. If you obtain a solution directly from the internet, you must cite the link and explain the solution in your own words to avoid plagarizing.

Note: As we had to change the grading scheme, it is especially important that you work individually. In particular you should *not take a solution from someone else*. **However, it is permissable to discuss a problem with other students so as you are learning together. But then you should wait at least one hour before writing down your solution and you should not use any written notes from conversations with other students. Just keep in mind that the goal is to understand the concepts. It is always important not to plagarize but if we are going to increase the weight of assignments, we clearly have to have some confidence that your work is your work.**

1. (20 points) Consider again the weighted set packing problem as in Assignment A3. The input is a collection $\mathcal{C} = \{S_1, S_2, \ldots, S_m\}$ of sets where $|S_i| \leq k$ for some fixed integer k and w_i is the weight of S_i. The objective is to select a subcollection $\mathcal{C}'$ of disjoint sets so as to maximize the objective function $F(\mathcal{C}') = \sum_{S_i \in \mathcal{C}'} w_i$.

   Consider the oblivious local search algorithm $LOCAL$ that for any current solution $\mathcal{C}'$, tries to add a set $S_i \in \mathcal{C} \setminus \mathcal{C}'$ (so as to improve the objective function F), removing any $S_j \in \mathcal{C}'$ that intersects S_i. The algorithm stops when there is no such $C_i \in \mathcal{C} \setminus \mathcal{C}'$ that improves F. Initially, we can set $\mathcal{C}' = \emptyset$ or S_i for any single set S_i. Show that $LOCAL$ achieves a k approximation ratio. That is, for every input $\mathcal{C}$, $LOCAL(\mathcal{C}) \geq \frac{1}{k} OPT(\mathcal{C})$. We suggest using a charging argument to show that your algorithm obtains the stated approximation.

2. (20 points) Consider the Exact Max-2-SAT problem and the oblivious local search algorithm with Hamming distance neighbourhood $N_1(\tau)$ as defined in the slides for week 10. This algorithm achieves approximation (totality) ratio $\frac{2}{3}$. Suppose we extend the neighbourhood of any truth assignment to $N_1'(\tau) = N_1(\tau) \cup \{\overline{\tau}\}$ where $\overline{\tau}$ is the complement truth assignment; that is, $\overline{\tau} = true$ iff $\tau = false$.

   Prove that the same oblivious local search algorithm using neighbourhood N_1' acheives an approximation (totality) ratio of $\frac{3}{4}$.

   Note: You can just consider the unweighted problem as the proof for the weikghted problem would be the same. You will receive reasonable credit if you can argue why this extended neighbourhood will help improve the totality ratio knowing what you know about the analysis for the N_1 neighbourhood.

3. (20 points) Consider the following weighted Max-Sat problem:
We are given a weighted CNF formula $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ where each clause C_i has a positive integer weight w_i. Furthermore, all the clauses in F have either 3 or 4 literals and that $\sum_{i:C_i \text{ has 3 literals}} w_i = \sum_{i:C_i \text{ has 4 literals}} w_i$.
The goal is to find a truth assignment so as to maximize the weight of satisfied clauses.

   Provide a polynomial time randomized algorithm whose objective is to maximize in expectation the weight of satisfied clauses. What guarantee can you provide on the expected weight of satisfied clauses. Explain your answer.

4. (20 points) Consider two polynomial size (i.e., number of aritmetic operations) arithmetic circuits C_1 and C_2 each having operations $+, -, \times$ and inputs $x_1, x_2, \ldots, x_n, c_1, \ldots c_m$ where the $c_i \leq n$ are constants in $\mathbb{N}$. Furthermore, the circuits have depth $O(\log n)$. The output of these circuits are polynomials $P_1(x_1, \ldots x_n)$ and $P_2(x_1, \ldots, x_n)$.

   We say that two such circuits are equivalent (denoted $C_1 \equiv C_2$) if P_1 and $P2$ are identical multivariate polynomials. For example, $(x_1 - x_2) \cdot (x_1 + x_2) \equiv x_1^2 - x_2^2$. Describe a 1-sided error randomized polynomial (in n) time algorithm ALG that will determine if $C_1 \equiv C_2$ or $C_1 \not\equiv C_2$ with high probability. Namely, ALG must satisfy the following:

   - If $C_1 \equiv C_2$, then ALG will say YES
   - If $C_1 \not\equiv C_2$, then ALG will say NO with probability at least $1 - 2^{-n}$.

5. (20 points) We have posted on the 373 web page, lecture notes by Dan Spielman (Yale University) on Schoning's algorithm for 3SAT. Those notes start with a slighlty different algorithm and somewhat simpler analysis resulting in a 1-sided error randomized $\tilde{O}(\frac{3}{2})^n)$ time bound (say with constant error probability) where $\tilde{O}$ hides logarithmic factors.
The slides then show how to obtain Schoning's algorithms obtaining time $\tilde{O}(\frac{4}{3})^n)$.

   In this exercise you are to adapt the simpler analysis for the 4-SAT problem. What is your time bound?