CSC 2420 Spring 2017, Assignment 3
Due date: April 17. Assignments submitted after April 19 will incur a penalty of 5% per additional day of lateness.

It is certainly preferable for you to solve the questions without consulting a published source. However, if you are using a published source then you must specify the source and you should try to improve upon the presentation of the result.

If you would like to discuss any questions with someone else that is fine BUT at the end of any collaboration you must spend at least one hour playing video games or watching two periods of Maple Leaf hockey or maybe even start reading a good novel before writing anything down.

Unless stated otherwise, all subquestions are worth 10 points. If you do not know how to answer a question, state "I do not know how to answer this (sub) question" and you will receive 20% (i.e. 2 of 10 points) for doing so. You can receive partial credit for any reasonable attempt to answer a question BUT no credit for arguments that make no sense.

In class I can clarify any questions you may have about this assignment.

1. Consider the following questions concerning the heavy hitters problem.

   (a) Provide a proof that if there is a majority element in a stream, then that value will be the value of *Candidate* as computed by the Majority algorithm.

   (b) Show how you could set ℓ and b in the count-min sktech so that the estimated count Z computed will satisfy the (ϵ, δ) result that:

   $$Prob[Z > c(x) + \epsilon n] \leq \delta$$

   . Can you find more than one setting for ℓ and b?
   Hint: First set b to obtain a bound on the expectation of Z_i and then set ℓ to obtain the stated (ϵ, δ) result.

2. The following questions relate to Lecture 12.

   (a) Verify that the objective functions for the max-k-densest subgraph problem and the max-k-vertex coverage problem have been properly formulated as quadratic programs. That is, when the variables are restricted to $\{-1, 1\}$, we obtain the optimal solution for the problem.

   (b) Formulate the max-k-dicut problem as a quadratic program with variables $\{-1, 1\}$.

   (c) In terms of $\{-1, 1\}$ variables explain why the triangle inqualiities hold.

3. The following questions relates to Lecture 8 and the discussion there of Schöning's k-SAT algorithm.

   (a) Schöning's k-SAT algorithm is based on the fact that if a random assignment τ starts of being Hamming distance j from an arbitrary satisfying assignment τ^*, then the probability that the random walk (as described in Lecture 8) will find τ^* (or another satisfying assignment) is at least $(\frac{1}{k-1})^j$. Complete the argument as to why this implies that the expected time find a satisfying assignment (assuming one exists) is $\tilde{O}[2(\frac{k-1}{k})]^n$. Note that this is $\tilde{O}(c^n)$ for $c < 2$ always beating the trivial $\tilde{O}(2^n)$ for each fixed k.

(b) Schöning further shows that the same idea will lead to a method for solving CSPs with n variables over a demain of d elements better than the trivial $\tilde{O}(d^n)$. Suppose we have a CSP where each "constraint" C depends on at most ℓ variables. Informally, he replaces 2^n by d^n and k by $\ell(d-1)$. State Schöning's algorithm for CSPs having at most ℓ literals per clause and domains of d elements whose time bound is $\tilde{O}(c^n)$ for some $c < d$ thereby improving upon the trivial d^n.

4. The following question relates to the time complexity for optimally coloring a 3-colorable graph (or determining if a graph is 3-colorable). Here the most naive algorithm would have time 3^n. Here we assume the graph is simple and connected.

   (a) Show that there is a backtracking algorithm that achieves time complexity $\tilde{O}(2^n)$.

   (b) Show that there is a backtracking algorithm that achieves time complexity $\tilde{O}(3^{n/3}) \approx (1.422)^n$. Hint: Use the fact that there are at most $O(3^{n/3})$ maximal indepedent sets in a graph with n nodes.

   (c) Show how to apply Schöning's CSP algorithm to this problem to achieve a $\tilde{O}(c^n)$ tme algorithm for $c < 2$. What is your time complexity? If the expected time complexity of this algorithm is not as good as the bound in part (b), why might we still want to use it.

   Aside: I think the best known time bound is $\approx (1.3289)^n$ to find a 3 coloring if one exists. Also much work on approximately coloring a 3 colorable graph.