Carnegie Mellon University Instructor: Venkatesan Guruswami
15-252: More GITCS Term: Spring 2021

Problem Set 12
Due date: Friday, May 7 (by midnight EDT)

Instructions

- You are allowed to collaborate with one other student taking the class, or do it solo.
- Collaboration is defined as discussion of the lecture material and solution approaches to the problems. Please note that *you are not allowed to share any written material and you must write up solutions on your own.* You must clearly acknowledge your collaborator in the write-up of your solutions.
- Solutions must be submitted on gradescope. Typesetting in LaTeX is recommended but not required. If submitting handwritten work, please make sure it is a legible and polished final draft.
- You should not search for solutions on the web. More generally, you should try and solve the problems without consulting any reference material other than what we cover in class and any provided notes.
- Please start working on the problem set early. Though it is short, the problem(s) might take some time to solve.

---

1. Let us consider the following natural randomized algorithm for k-SAT, the satisfiability problem for formulae in Conjunctive Normal Form (CNF) with clauses of width at most k.

   Algorithm: On input a k-SAT instance φ on n variables $x_1, x_2, \ldots, x_n$:

   (a) Pick a random ordering $\sigma_1, \sigma_2, \ldots, \sigma_n$ of the n variables of the formula φ

   (b) For $i = 1$ to n:

   i. If σ_i (or its negation) is present in a unit clause (i.e., that has a single literal), then set σ_i to satisfy that clause.

   ii. Else set σ_i to true or false uniformly at random.

   iii. Simplify the formula based on the assignment to σ_i

   (c) If all clauses are satisfied by the assignment, then output the assignment, else output fail.

   It turns out that if φ is satisfiable, then this algorithm succeeds in finding a satisfying assignment for φ with probability at least $\frac{1}{2n}2^{-n+n/k}$. Repeating the algorithm enough times then gives a $2^{n-n/k}\text{poly}(n)$ time randomized algorithm for k-SAT with high success probability.

   In this exercise, your goal is to prove this in the special case when φ has a unique satisfying assignment. Let $\bar{a}$ denote this unique (unknown) satisfying assignment.

(a) Prove that for every variable x_i there is a clause C_i of φ containing $\ell_i \in \{x_i, \overline{x_i}\}$ such that ℓ_i is the only literal of C_i that is set to true under $\bar{a}$.

For each x_i, we call such a C_i its *identifying clause* (if there is more than one possible choice for C_i, choose an arbitrary one as the identifying clause). Note that the identifying clauses of distinct variables are distinct.

(b) Call a variable x_i *good* w.r.t an ordering $\sigma_1, \sigma_2, \dots, \sigma_n$ if it occurs last amongst the variables in its identifying clause under this order.

Show that the expected number of variables that are good w.r.t a random ordering $\sigma_1, \sigma_2, \dots, \sigma_n$ is at least $\frac{n}{k}$.

(c) Call an ordering *groovy* if there are at least $\frac{n}{k} - 1$ variables that are good w.r.t it. Show that a random ordering is groovy with probability at least $\frac{1}{n}$.

(d) Prove that if the algorithm picks a groovy ordering in the first step, it will then find and output $\bar{a}$ with probability at least $2^{-n+n/k-1}$.

(e) Conclude that the above algorithm finds the unique satisfying assignment $\bar{a}$ with probability at least $\frac{1}{2n}2^{-n+n/k}$.