15-859FF: Coping with Intractability **CMU, Fall 2019**
HW #2 (out Saturday Sep 28, 2019) **Due:** Sunday Oct 13, 2019

You may discuss all problems on this HW with others, but groups of size at most 3, please. Submission details (and corrections) will appear on Piazza, please check it regularly.

Exercises

Exercises are for fun and edification, please do not submit. **But do solve them, we may need ideas from there later in the course, or even in this HW!**

1. Suppose you are given two sorted lists A and B of n numbers each. Given a number c, you want to find $a \in A, b \in B$, such that $a + b = c$ (or report none exist). Give an algorithm for this problem that takes time $O(n \log n)$. Improve this to an algorithm that takes time $O(n)$. Show that you cannot do better than $\Theta(n)$ time.

2. Suppose you are given as input an assignment a which is promised to be within Hamming distance d from a satisfying assignment to a k-SAT formula ϕ with n variables. Extend the argument from lecture to give a deterministic algorithm that finds a satifsying assignment to ϕ in $k^d\text{poly}(n)$ time.

3. In the *Longest Non-Decreasing Subsequence* problem, we are given a sequence $A = a_1 a_2 \ldots a_n$ of integers, and the goal is to delete the fewest integers a_i such that the remaining subsequence is non-decreasing. E.g., you can convert 005311231525 into $00 - -1123 - 5 - 5$ by deleting four elements. Solve this problem in polynomial time.

4. You perform an unbiased random walk on the integers, starting at zero and stopping when you get to $+n$ or $-n$ (the boundaries). If $T(i)$ is the time to reach the boundaries starting at i, write down a recurrence for expected time to reach the boundaries, and show that $T(0) = n^2$.

   Another approach that gives a loose lower bound: if $X_i \in \{-1, +1\}$ u.a.r. (uniformly at random) and X_i, X_j are independent for $i \neq j$, and $S_T = \sum_{i=1}^{T} X_i$, then show that $\mathbf{E}[S_T] = 0$ and $\mathbf{Var}(S_T) = T$. Use Chebyshev's inequality to show that $\mathbf{Pr}[|S_T| \geq c\sqrt{T}] \leq \frac{1}{c^2}$. Hence infer that the expected time to reach the boundaries is at least $n^2/4$.

5. Let ϕ be a 3SAT instance with n variables, and suppose you are given a maximal collection of t clauses that are disjoint (they do not share any variables, and you cannot add any other clause that is also disjoint with these t clauses). Find a satisfying assignment of ϕ in $O^*(7^t)$ time.

6. It is known that FVS has a 2-approximation algorithm in polynomial time. Suppose we are given a FVS of size $2k$. Show that Disjoint FVS can be solved in time $25^k(m + n)$. (This algorithm has the benefit of being linear in m and n, provided that the 2-approximation algorithm is also linear.)

Problems

The Cygan et al book provides hints to its problems. Please try to solve these problems yourself before looking at these hints! Recall the notation $O^*(c^n)$ to denote $O(c^n\text{poly}(n))$; this helps avoid repeating polynomial factors in many places below. Do #5 and three out of the other four?

1. (5.8)

2. In the induced subgraph isomorphism, given G and H, we wanted to find a subset $S \subseteq V$ of vertices that induce a copy of H, i.e., such that $G[S]$ is isomorphic to H. We consider the case where G has bounded degree d. We saw in lecture (and in a piazza post) that when H is connected, we can delete each edge of G with probability $1/2$, and then find a copy of H among the resulting components (if one exists). This succeeds with probability 2^{-dk}, where $k = |V(H)|$.

   Suppose H is not connected, this does not work. (Why?) Give a color-coding algorithm to solve induced subgraph isomorphism in time $O^*(2^{(d+1)k}k!)$ time.

3. (10.8)

4. In the Exact-One-SAT problem, we are given as input a CNF formula ϕ, and the goal is to determine if there is a Boolean assignment to the variables which sets *exactly one literal* true in each clause. Give an algorithm that runs in time $O^*(2^{n/2})$ time to solve the Exact-One-SAT problem, where n is the number of variables. (Hint: solve exercise #1.)

5. In this problem, we will consider the following simple randomized algorithm for k-SAT.

Algorithm: On input a k-CNF formula ϕ on n variables:

(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.

Our goal is to prove that the above algorithm finds a satisfying assignment (if one exists) with probability at least $\approx 2^{-n+n/k}$. Then, of course, by repeating the trial $O^*(2^{n-n/k})$ times, we get an algorithm that succeeds with high probability.

(a) Suppose that x is a satisfying assignment to ϕ, and consider the n assignments $x^{(j)}$ formed by flipping (only) the j'th variable in x (in some canonical order). Let $b(x)$ be the number of these assignments that also satisfy ϕ (so $b(x)$ is an integer in the range $[0, n]$). Then prove that the above algorithm outputs x with probability at least $2^{-n+(n-b(x))/k} n^{-O(1)}$.

(b) The following fact is true for all satisfiable ϕ:

$$\sum_{x:x \text{ satisfies } \phi} 2^{-b(x)} \geq 1 \ .$$

(You don't have to prove this fact, but it is a nice exercise; induction is one approach.) Using the above and Part (5a), prove that the algorithm outputs a satisfying assignment of ϕ with probability at least $2^{-n+n/k} \cdot n^{-O(1)}$.