CS120: Intro. to Algorithms and their Limitations	Hesterberg & Anshu
Problem Set 8	
Harvard SEAS - Fall 2023	*Due: Wed Nov. 29, 2023 (11:59pm)*

Your name:
Collaborators:
No. of late days used on previous psets:
No. of late days used after including this pset:
Please fill out feedback form: https://forms.gle/K4Z1b1EhsT8dRY2T6.

The purpose of this problem set is to to reinforce the definitions of $\mathsf{P}_{\mathsf{search}}$, $\mathsf{EXP}_{\mathsf{search}}$, $\mathsf{NP}_{\mathsf{search}}$, and $\mathsf{NP}_{\mathsf{search}}$-completeness and practice NP-completeness proofs.

1. (Positive Monotone SAT) A boolean formula is *positive monotone* if there are no negations in it. Restricting SAT to positive monotone formulas makes it a trivial problem: setting all variables to 1 is always a satisfying assignment.

   However, the following variant of Positive Monotone SAT is more interesting:

Input	: A positive monotone CNF formula $\varphi(x_0, \ldots, x_{n-1})$ and a number $k \in \mathbb{N}$
Output	: A satisfying assignment $\alpha \in \{0,1\}^n$ in which at least k variables are set to 0 (if one exists)

   Computational Problem k-False PositiveMonotoneSAT

   (a) Prove that k-False PositiveMonotoneSAT is $\mathsf{NP}_{\mathsf{search}}$-complete, even when $k = n/2$. (Hint: reduce from SAT, replacing negated variables with new ones and adding additional clauses.)

   (b) Prove that if we fix $k = 3$, then k-False PositiveMonotoneSAT is in $\mathsf{P}_{\mathsf{search}}$. (Hint: show that it suffices to consider assignments in which exactly 3 variables are set to 0.)

   (c) (optional[1]) Show that k-False PositiveMonotone 2-SAT is $\mathsf{NP}_{\mathsf{search}}$-complete. (Hint: reduce from Independent Set.)

2. (Reductions and complexity classes)

   (a) Prove that if a problem Π is in $\mathsf{P}_{\mathsf{search}}$, then $\Pi \leq_p \Gamma$ for all computational problems Γ.

   (b) Show that if $\mathsf{NP}_{\mathsf{search}} \subseteq \mathsf{P}_{\mathsf{search}}$, then all problems in $\mathsf{NP}_{\mathsf{search}}$ are $\mathsf{NP}_{\mathsf{search}}$-complete. (The converse of this statement was proved in section, so it is actually an iff.)

   (c) Prove that if $\Pi \leq_p \Gamma$ and $\Gamma \in \mathsf{EXP}_{\mathsf{search}}$, then $\Pi \in \mathsf{EXP}_{\mathsf{search}}$. (In other words, $\mathsf{EXP}_{\mathsf{search}}$ is closed under polynomial-time reductions.)

[1] This problem is meant to be done based on your enjoyment/interest and only if you have time. It won't make a difference between N, L, R-, and R grades, and course staff will deprioritize questions about this problem at office hours and on Ed.

3. (Variant of VectorSubsetSum) In the Sender–Receiver Exercise on November 16, you will see that the following problem is $\mathsf{NP}_{\mathsf{search}}$-complete.

Input	: Vectors $\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_{n-1} \in \{0,1\}^d$, $\vec{t} \in \mathbb{N}^d$
Output	: A subset $S \subseteq [n]$ such that $\sum_{i \in S} \vec{v}_i = \vec{t}$, if such a subset S exists.

Computational Problem VectorSubsetSum

We will use the notation $\vec{v}[j]$ to denote the j'th entry of vector $\vec{v}$, so the condition $\sum_{i \in S} \vec{v}_i = \vec{t}$ means that for every $j = 0, 1, \ldots, d-1$, we have $\sum_{i \in S} \vec{v}_i[j] = \vec{t}[j]$.

Assuming that result, prove that the following variant is also $\mathsf{NP}_{\mathsf{search}}$-complete.

Input	: Vectors $\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_{n-1} \in \mathbb{N}^d$, $t_0 \in \mathbb{N}$
Output	: A subset $S \subseteq [n]$ such that $\sum_{i \in S} \vec{v}_i = (t_0, t_0, \ldots, t_0)$, if such a subset S exists.

Computational Problem VectorSubsetSumVariant

The two differences from the VectorSubsetSum problem is that the vectors are no longer restricted to have $\{0,1\}$ entries, but now all entries of the target vector are required to be equal. (Hint: reduce from the standard VectorSubsetSum problem. Add an additional vector and an additional coordinate.)