Exercise IX, Theory of Computation 2025

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 A k-CNF formula is a CNF formula where each clause has at most k literals. The language $k\mathsf{SAT}$ consists of all k-CNF formulas which are satisfiable. What is wrong with the following proof of **NP**-completeness of 2SAT?

Since 3SAT *is in* **NP***, so is* 2SAT*. For any 2-CNF* $\varphi = (a_1 \vee b_1) \wedge \cdots \wedge (a_n \vee b_n)$*, define the 3-CNF* $f(\varphi) = (a_1 \vee b_1 \vee b_1) \wedge \cdots \wedge (a_n \vee b_n \vee b_n)$*, where the* a_i *and* b_i *are literals. Note that*

$$\varphi \text{ is satisfiable} \iff f(\varphi) \text{ is satisfiable.}$$

Hence $3\mathsf{SAT} \leq_P 2\mathsf{SAT}$*, and it follows that* 2SAT *is* **NP***-complete.*

Solution: The problem with the proof is that the function φ goes from 2SAT to 3SAT. This only proves $2\mathsf{SAT} \leq_p 3\mathsf{SAT}$, which while true, is not the desired direction. To prove that 2SAT is **NP** complete, we would need a function that maps any given 3SAT instance to a 2SAT instance.

2 Prove that the following problem is **NP**-complete: Given a set S, a collection $\mathcal{C}$ of subsets of S and an integer k, is there a subset $T \subseteq S$ of size at most k such that $T \cap C \neq \emptyset$ for all $C \in \mathcal{C}$?

Solution: This problem is called HittingSet. To prove it is **NP**-complete, we will first show that it is in **NP**. After that, we will show HittingSetis **NP**-hard by reducing the vertex cover problem VC to it.

An instance of HittingSetis a triple $(S, \mathcal{C}, k)$, and as a witness we can take the set T. The verifier V can be constructed as follows.

On input $((S, \mathcal{C}, k), T)$, do:

1. If $|T| > k$, reject.
2. If $T \not\subseteq S$, reject.
3. For every $C \in \mathcal{C}$, do:
   (a) If $T \cap C = \emptyset$, reject.
4. Accept.

We observe that V runs in polynomial time, and $V((S, \mathcal{C}, k), T)$ accepts if and only if $(S, \mathcal{C}, k)$ is a yes instance of our problem.

It remains to show that HittingSetis **NP**-hard by reducing from the vertex cover problem VC. First, we recall that an instance $(G = (V, E), k)$ is in VC if and only if there is a subset S of V with at most k vertices, such that every edge in E has at least one vertex in S.

We define the reduction function f as follows.

On input $((V, E), k)$, do:

1. Define $\mathcal{C} = \{\{e\} : e \in E\}$.
2. Output $(V, \mathcal{C}, k)$.

Note that f is polynomial-time computable. We now show that $((V, E), k) \in \mathsf{VC}$ if and only if $f((V, E), k) = (V, \mathcal{C}, k) \in$ HittingSet.

- Assume that $(G, k) \in \mathsf{VC}$: Then G has a vertex cover T of size at most k. By definition, T is also a hitting set for $(V, \mathcal{C}, k)$ since for each set $C \in \mathcal{C}$, $C \cap T \neq \emptyset$.

- Assume that $(V, \mathcal{C}, k)$ has a hitting set T: Then by definition, T is a vertex cover for (G, k).

3 Using the **NP**-completeness SubsetSum, prove that the following problem is **NP**-complete: Given integers $V, v_1, \ldots, v_n$ and $W, w_1, \ldots, w_n$, is there a subset S of $\{1, 2, \ldots, n\}$ such that

$$\sum_{i \in S} w_i \leq W \text{ and } \sum_{i \in S} v_i \geq V?$$

Solution: Let's denote this problem as KnapSack. We first recall that an instance to the SubsetSum problem consists of a set of integers S and an integers M. $(S, M) \in$ SubsetSum if and only if there is some subset T of S such that the sum of elements of T is M.

We now show that KnapSack $\in$ **NP**. An instance of KnapSackis $((V, v_1, \ldots, v_n), (W, w_1, \ldots, w_n))$ and as a witness we can take the set S. Define the verifier as follows.

On input $(((V, v_1, \ldots, v_n), (W, w_1, \ldots, w_n)), S)$, do:

1. If $S \not\subseteq [n]$, reject.
2. Compute $s_1 = \sum_{i \in S} w_i$.
3. Compute $s_2 = \sum_{i \in S} v_i$.
4. Accept if $s_1 \leq W$ and $s_2 \geq V$. Otherwise, reject.

Observe that this algorithm runs in polynomial time. Moreover, it accepts if and only if the instance is a yes-instance.

It remains to show that KnapSackis **NP**-hard. We do so by reducing from SubsetSum using f defined as follows.

On input (S, M), do:

1. Let $S = \{s_1, \ldots, s_n\}$.
2. Output $((M, s_1, \ldots, s_n), (M, s_1, \ldots, s_n))$.

It is clear that f runs in polynomial time. We now show that $(S, M) \in$ SubsetSum if and only if $f(S, M) \in$ KnapSack.

- Assume that $(S, M) \in$ SubsetSum: Then there exists some $T \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in T} s_i = M$. Therefore, using the same set T as a witness, $f(S, M) \in$ KnapSack.

- Assume that $f(S, M) \in$ KnapSack: Then there exists some $T \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in T} s_i \geq M$ and $\sum_{i \in T} s_i \leq M$, which implies that $\sum_{i \in T} s_i = M$. Thus, $(S, M) \in$ SubsetSum, as desired.

4* Prove that the following problem is **NP**-hard: Given vectors $v_1, \dots, v_m, V \in \mathbb{Z}^n$ and $K \in \mathbb{Z}$, do there exist $a_1, \dots, a_m \in \mathbb{Z}$ such that

$$\left\| \sum_{i=1}^{m} a_i v_i - V \right\|^2 \leq K?$$

Note that for a vector $v \in \mathbb{R}^n$, we have $\|v\|^2 = \sum_{i=1}^n v_i^2$.

Hint: *First consider the variant of the problem where $a_1, a_2, \dots, a_m \in \{0, 1\}$.*

Solution: Let us denote this problem by ClosestVector. We show that ClosestVectoris **NP**-hard by reducing from SubsetSum. We first explain the intuition behind the reduction.

Note that the problem SubsetSum can be reformulated as finding $a_1, \dots, a_n \in \{0, 1\}$ such that $\sum_{i=1}^n a_i s_i = M$, where the integers s_i and M are given. Now a natural idea would be to take $v_i = s_i v$, $V = Mv$ and $K = 0$ for some fixed vector v. Then if $(S, M) \in$ SubsetSum, there are $a_1, \dots, a_n \in \{0, 1\}$ such that

$$\left\| \sum_{i=1}^{n} a_i v_i - V \right\|^2 = \left| \sum_{i=1}^{n} a_i s_i - M \right|^2 \|v\|^2 = 0 \leq K.$$

However, it is also possible that $(S, M) \notin$ SubsetSum but still $\| \sum_{i=1}^n a_i x_i - M \|^2 = 0$ for some other values of the a_i, which need not be in $\{0, 1\}$. Consider $S = \{1\}$, $M = 2$, with $(S, M) \notin$ SubsetSum, but if $a_1 = 2$ we still get $|1 \cdot a_1 - 2| = 0$. Thus, we need to modify the reduction to make sure $a_i \in \{0, 1\}$ for all i. Consider the function f defined as follows.

On input (S, M), do:

1. For $i \in \{1, 2, \dots, n\}$, do:
   (a) Initialise $v_i = 0 \in \mathbb{Z}^{n+1}$.
   (b) Set $(v_i)_1 = s_i$.
   (c) Set $(v_i)_{i+1} = 2$.
2. Set $V = (M, 1, \dots, 1) \in \mathbb{Z}^{n+1}$ and $K = n$.
3. Output $((v_1, \dots, v_n), V, K)$.

Clearly f runs in polynomial time. We show that $(S, M) \in$ SubsetSum if and only if $f(S, M) = ((v_1, \dots, v_n), V, K) \in$ ClosestVector.

- Assume that $(S, M) \in$ SubsetSum: There exists $T \subseteq S$ such that $\sum_{s \in T} s = M$. For all i, let $a_i = 1$ if $s_i \in T$ and $a_i = 0$ otherwise. Let $u = \sum_{i=1}^n a_i v_i$. By construction, we know that $u_1 = M$ and $u_i = 0$ or 2 for $2 \leq i \leq n + 1$. Thus, V and u differ by 1 in exactly n coordinates. Thus, $\| \sum_{i=1}^n a_i v_i - V \|^2 = \sum_{i=1}^n (\pm 1)^2 = n \leq K$ as desired.

- Assume that $f(S, M) \in$ ClosestVector: There exists $a_1, \dots, a_n \in \mathbb{Z}$ which satisfy $\| \sum_{i=1}^n a_i v_i - V \|^2 \leq n$. Again set $u = \sum_{i=1}^n a_i v_i$. Observe that for all $2 \leq i \leq n + 1$, u_i is even but $V_i = 1$. Thus, u and V differ by at least 1 in at least n coordinates, which implies $\| \sum_{i=1}^n a_i v_i - V \|^2 \geq n$. Therefore, we must have $\| \sum_{i=1}^n a_i v_i - V \|^2 = n$. This implies that we must have equality in the last coordinate, $u_1 = V_1 = M$, and all the a_i are either 0 or 1. Let T be the set of s_i where $a_i = 1$. Since $u_1 = M$, This set T now witnesses the fact that $(S, M) \in$ SubsetSum, as desired.