CS513 HW3 Sample Solutions

Notice: you can use any known NP-hard problems for the polynomial reduction.

1. In a BoxTiling problem, there is a large Box R (in 3D) with width W, length L and height H, and several small boxes $r_1, r_2, \cdots, r_n$ where box r_i has width w_i, length ℓ_i and height h_i. The question is whether there is a subset of the small boxes that can be put inside the large box R with no gaps or overlaps. Show that BoxTiling is NP-hard.

   Solution: Reduction from SubsetSum (given numbers $x_1, x_2, \cdots x_n$, is there is a subset of numbers that add up to $X = \sum_i x_i/2$). We create boxes with width 1, length 1 and height x_i. And make R to be width 1, length 1 and height X. If we answer yes to SubsetSum then we answer yes to BoxTiling.

2. Adam wants to ask if there is a simple path (i.e., not repeating vertices) in a graph G that goes through at least 1/3 of the vertices. Show that this problem is NP-hard.

   Solution: Reduce from Hamiltonian Path problem. Given a graph G of n vertices, create $2n$ additional singleton vertices that are not connected to any other vertices. Call this new graph G'. The existence of a hamiltonian path in G is equivalent to the existence of a path in G' that visits 1/3 vertices.

3. Consider a directed graph $G = (V, E)$, a number k, and a set of paths $P_1, P_2, \cdots P_m$ of G, is it possible to select at least k of the paths such that no two of the selected paths share any vertices? Show that this problem is NP-hard.

   Solution: We reduce from 3D matching to this problem. Given Y_1, Y_2, Y_3 each of n vertices and and 3-tuples $U = (u, v, w)$ with $u \in Y_1$, $v \in Y_2$, and $w \in Y_3$, we ask whether there are n tuples of U such that each vertex of Y_1, Y_2, Y_3 appears exactly once. Now we formulate the graph in our problem. The vertices are $Y_1 \cup Y_2 \cup Y_3$ and the edges are the tuples in U. We take $k = n$. If 3D matching answers YES, then there is a subset of k paths that are disjoint. The opposite direction is also true.

4. Given a finite set U of size n and a collection $A_1, A_2, \cdots A_m$ of subsets of U. You are alkso given numbers $c_1, c_2, \cdots c_m$. The question is that, does there exist a subset $X \subseteq U$ such that the cardinality of $X \cap A_i$ is equal to c_i? Prove that this problem is NP-complete.

   (Hint: you may use some variants of 3SAT: `https://en.wikipedia.org/wiki/Boolean_satisfiability_problem#3-satisfiability`)

   Solution: (Updated December 13, 2025; correction credited to Jason Liu.)

   The problem is in NP. Given a candidate subset $X \subseteq U$, we can compute $|X \cap S|$ for every input set S and check whether it equals the prescribed number. This verification runs in time polynomial in the input size.

NP-hardness. We reduce from one-in-three 3-SAT, where exactly one literal in each clause is TRUE. Given an instance with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_k$, define

$$U = \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}.$$

For each clause $C_i = (\ell_{i1} \vee \ell_{i2} \vee \ell_{i3})$, define

$$A_i = \{\ell_{i1}, \ell_{i2}, \ell_{i3}\}, \qquad c_i = 1 \qquad (i = 1, \ldots, k).$$

For each variable x_j, define a consistency set

$$B_j = \{x_j, \bar{x}_j\}, \qquad d_j = 1 \qquad (j = 1, \ldots, n).$$

We output the instance of the set-intersection problem whose collection of subsets is $\{A_1, \ldots, A_k\} \cup \{B_1, \ldots, B_n\}$ with target numbers $\{c_1, \ldots, c_k\} \cup \{d_1, \ldots, d_n\}$.

If the one-in-three 3-SAT instance is satisfiable, let X be the set of all literals assigned TRUE. Then $|X \cap A_i| = 1$ for every clause i, and $|X \cap B_j| = 1$ for every variable j. Conversely, suppose there exists $X \subseteq U$ such that $|X \cap A_i| = 1$ for all i and $|X \cap B_j| = 1$ for all j. The constraints $|X \cap B_j| = 1$ induce a consistent truth assignment (choose x_j TRUE iff $x_j \in X$), and the constraints $|X \cap A_i| = 1$ imply that each clause has exactly one TRUE literal. Hence the one-in-three 3-SAT instance is satisfiable.

Therefore the problem is NP-complete.