1. **Approximation Set Cover:** An instance of the *Set Cover problem* consists of a finite ground set X and a family $\mathcal{A}$ of subsets of X, such that every element of X belongs to at least one subset in $\mathcal{A}$. We want to find a minimum-size subset $\mathcal{C} \subseteq \mathcal{A}$ whose members *covers* all elements of X.

   Consider the special case of Set Cover, where every element of X appears in at most k subsets of $\mathcal{A}$. Find a k-approximation algorithm for this case of Set Cover. That is, find a collection of covering sets that has size at most k times that of the optimal cover.

2. **SAT:** Model the *Satisfiability problem* with An Integer Linear Program.

3. **3-Coloring:** Model the 3-*Coloring problem* with an Integer Linear Program. The 3-Coloring problem is: Given a graph, is it possible to color each node either *red, green, or blue*, so that no two nodes of the same color have an edge between them?

4. **Hitting Set:** In the Hitting Set Problem, we have a set $A = \{a_1, a_2, \ldots, a_n\}$ and a collection $B_1, B_2, \ldots, B_m$ of subsets of A. We say that a set $H \subseteq A$ is a *hitting set* for the collection $B_1, B_2, \ldots, B_m$ if H contains at least one element from each B_i (that is, $\forall i, H \cap B_i \neq \emptyset$).

   Given an instance of this problem, we would like to determine whether there is a hitting set of size at most k for the collection. Further, suppose that each B_o has at most c elements, where c is a small constant. Given an algorithm to solve this problem with a running time of the form $O(f(c,k) \cdot p(n,m))$ where $p(\cdot)$ is a polynomial function of n and m, and $f(\cdot)$ is any function that depends only on c and k, not on n or m.

5. **Approximation Knapsack:** An instance of *Knapsack problem* consists of:

   - Items $I = \{1, \ldots, n\}$.
   - Sizes $s_1, \ldots, s_n$ for each of the corresponding items.
   - Profits $p_1, \ldots, p_n$ for each of the corresponding items.
   - Knapsack capacity B.

   A feasible solution to the Knapsack problem is a subset $I' \subseteq I$ such that $\sum_{i \in I'} s_i \leq B$. We want to maximize the profit $f(I') = \sum_{i \in I'} p_i$.

   Give an 2-Approximation algorithm to this problem. That is, if the optimal solution has profit P', then your algorithm should get profit $\geq P'/2$. Your algorithm should run in polynomial time.