(1)
The Hitting Set problem is defined as follows: given a set V of nodes with weights
wi for i ∈ V , and a list of subsets H whose elements are subsets of V , and an integer k. A hitting set
S is a subset S ⊂ V such that S ∩ H ≠ ∅ for all sets H ∈ H. The minimum cost hitting set

problem is to find a hitting set S of minimum total weight $\sum_{i \in S} wi$
Note that when all sets H ∈ H have exactly 2 elements, then the sets in H form the edges in a graph
G, and the hitting set problem is the vertex cover problem on the graph G. (This also shows that the Hitting Set problem is NP-hard).

a. For the special case of Vertex Cover we have seen a 2-approximation algorithm. Show that the bound of 2 we gave comparing the true optimum and the optimum for the
value of the linear program used in the approximation algorithm is best possible. Concretely, show that for any constant γ < 2 there is a vertex cover problem with weights wi such that the linear programming optimum is more than a factor of γ smaller than the true optimum.

b. Now consider the general case of the Hitting Set, and assume that the maximum size
of a set of H is maxh∈H |h| ≤ c. Give a c-approximation algorithm for this case of the min-weight
Hitting Set problem.

(2)
We have studied the approximation algorithms for the knapsack problem: the problem
had n items with weight wi and value vi, and weight limit W . The goal is to select a subset I ⊂ {1, . . . , n}

such that $\sum_{i \in I} wi \leq W$

and maximizing the total value $\sum_{i \in I} vi$
Here we want to consider a different
approximation: ensuring the total value of selected items is greater than the maximum value possible
with weight limit W , by allowing the total weight of the selected items to go somewhat beyond W .
a. Assume that no item is too big, that is wi ≤ W/2 for all i. Using this assumption,
give a polynomial-time algorithm that finds a solution with the following guarantee: Suppose the maximum value possible with weight limit W is V ∗, then the algorithm needs to output a subset

I of items, such that $\sum_{i \in I} wi \leq (3/2)W$ and $\sum_{i \in I} vi \geq V$
b. Give an algorithm with the guarantee in part (a) even if the input has big items.

(3) You are asked to create a schedule for the upcoming season of a sports league. Based on records from the previous year, it was decided which pairs of teams should play against each other in the upcoming season. Games are played on Sundays and each team can play at most one game on the same Sunday. The goal is to schedule all games in such a way that the number of weeks is minimized.

This scheduling problem is equivalent to the following problem on graphs (called edge coloring): You are given a graph G with vertices $t1, \ldots, tn$ corresponding to the teams. The edges of the graph correspond to the set of games that are to be scheduled in the upcoming season. An edge coloring of G assigns to each edge (ti, tj) in the graph a "color" wk such that no vertex is incident to more than one edge of the same color. (This constraint ensured that all games (ti, tj) with the same color wk can be held in the same week.) Your goal is to find an edge coloring of G that uses as few colors as possible.

Give an efficient algorithm to compute an edge coloring of G that uses at most $2d - 1$ different colors, where d is the maximum degree in the graph.

Further, show that the approximation ratio of this algorithm is at most 2.