I collaborated with: went to TA sessions

Question
Chapter 10, Problem 1. Hint: Try the idea we used for vertex cover (see Property 10.3): Delete some element of some B_i (and B_i) from the problem instance Think about how this reduction to a smaller problem can help.
Solution
Let $I = \{B_1, ..., B_m\}$ be the collection of subsets of the set A and k be the maximum size of the Hitting Set. Our base cases are if I is empty then we return true, and if $k == 0$ then we return false. Let $b_i \in B_i$. We remove every element of I that contains b_i and decrease k by one. This forms the recursive call. Where the parameters are the collection of subsets I and k. We do this for all c elements of every B_i. If at any point true is returned then there exists a hitting set of size at most k. However, if false is returned then no such set exists.
Time Complexity: There are at most c recursive calls for each value of k, so c^k nodes. Removing sets require visiting at most m sets and checking at most c elements in every set. So $O(c^k * cm) = O(m)$.

Proof. Base Case: If there are no subsets of A, then our Hitting Set has hit all elements of I. Also, if $k = 0$ then this means that there are still subsets in I not being hit by our "Hitting Set."
Inductive Hypothesis: Assume there is a hitting set of size $k - 1$ for a collection of subsets of A.
Inductive Step: Let H be such hitting set of size $k - 1$ for a collection of subsets of A. If we add an element to H then it also hits subsets of A containing this element. Thus, H is still a hitting set. Now, suppose H was not a hitting set for a collection of subsets. Then, even if we add an element to H and add subsets containing said elements, H will not be a hitting set because the original subsets are still not hit by the elements of H. □