Our solution will be similar to the algorithm for VERTEX COVER from the first section of this chapter. Consider the notation defined in the problem. For an element $a \in A$, we *reduce the instance by* a by deleting a from A, and deleting all sets B_i that contain a. Thus, reducing the instance by a producing a new, presumably smaller, instance of HITTING SET.

We observe the following fact. Let $B_i = \{x_1, \ldots, x_c\} \subseteq A$ be any of the given sets in the HITTING SET instance. Then at least one of $x_1, \ldots, x_c$ must belong to any hitting set H. So by analogy with (2.3) from the notes, we have the following fact

- Let $B_i = \{x_1, \ldots, x_c\}$ There is k-element hitting set for the original instance if and only if, for some $i = 1, \ldots, c$, the instance reduced by x_i has a $(k-1)$-element hitting set.

The proof is completely analogous to that of (2.3). If H is a k-element hitting set, then some $x_i \in H$, and so $H - \{x_i\}$ is a $(k-1)$-element hitting set for the instance reduced by x_i. Conversely, if the instance reduced by x_i has a $(k-1)$-element hitting set H', then $H' \cup \{x_i\}$ is a k-element hitting set for the original instance.

Thus, our algorithm is as follows. We pick any set $B_i = \{x_1, \ldots, x_c\}$. For each x_i, we recursively test if the instance reduced by x_i has a $(k-1)$-element hitting set. We return "yes" if and only if the answer to one of these recursive calls is "yes." Our running time satisfies $T(m,k) \leq cT(m,k-1) + O(cm)$, and so it satisfies $T(m,k) = O(c^k \cdot kcm)$. This gives the desired bound, with $f(c,k) = kc^{k+1}$ and $p(m) = m$.

[1] ex579.588.787