

$$h(k, i) = (h_1(k) + pRand(k, i)) \bmod m$$

After racking my brain for hours on end, with very limited resources on how to answer a question of this nature, I concluded that a modification of double hashing meets all the requirements to answer it. If there exists another approach to answering this question, I don't believe I will ever find it.

`pRand()` refers to a function that pseudo-randomly picks an 'i' such that the given 'i' has not been chosen previously during the insertion, and $0 \leq i < m$. Because it is pseudo random and uses *k* as a seed, we know it will be the same every time it's hashed. Since we know it is forced to select from the range $0 \leq i < m$ and that it cannot select something that has already been selected, we also know that it will search the entire hash table. I think its similarity to double hashing speaks for its efficiency which is why I designed it to be so similar to it.