Open Addressing: Linear Probing

3 min

Another popular hash collision strategy is called *open addressing*. In open addressing we stick to the array as our underlying data structure, but we continue looking for a new index to save our data if the first result of our hash function has a different key's data.

A common open method of open addressing is called *probing*. Probing means continuing to find new array indices in a fixed sequence until an empty index is found.

Suppose we want to associate famous horses with their owners. We want our first key, "Bucephalus", to store our first value, "Alexander the Great". Our hash function returns an array index 3 and so we save "Alexander the Great", along with our key "Bucephalus", into the array at index 3.

After that, we want to store "Seabiscuit"s owner "Charles Howard". Unfortunately "Seabiscuit" also has a hash value of 3. Our probing method adds one to the hash value and tells us to continue looking at index 4. Since index 4 is open we store "Charles Howard" into the array at index 4. Because "Seabiscuit" has a hash of 3 but "Charles Howard" is located at index 4, we must also save "Seabiscuit" into the array at that index.

When we attempt to look up "Seabiscuit" in our Horse Owner's Hash Map, we first check the array at index 3. Upon noticing that our key (Seabiscuit) is different from the key sitting in index 3 (Bucephalus), we realize that this can't be the value we were looking for at all. Only by continuing to the next index do we check the key and notice that at index 4 our key matches the key saved into the index 4 bucket. Realizing that index 4 has the key "Seabiscuit" means we can retrieve the information at that location, Seabiscuit's owner's name: Charles Howard.

Instructions

If we add a third key, "Secretariat", which also hashes to 3, where would the value be saved? What if we want to add a fourth key, "Epona" which returns from our hash function with an index of 5? When do you think we would use probing over other strategies?

In the workspace we've given an example of a hash map with a linear probing implementation. Some of the underlying array values of the hash map are already taken. Investigate where a key with a given hash code would be saved for different probing sequences.