Maps

The idea of a *map* is one of the fundamental concepts of programming. A map is a data structure that ties values to keys. Python's dictionaries are a typical map structure. If you give a map a key it gives you back the data associated with that key.

For example, we might have a map Ages in which the keys are names of people and the value associated with a key is that person's age.

We might have the following data: Michael is 21, Fernanda is 18, Samantha is 18, Aiesha is 30, George is 30, Peng is 21, Zejian is 19.

One way we might store this data is with an ArrayList<Entry>, where Entry is a simple class wit two fields: String key (or name) and int age. The map might look like this:

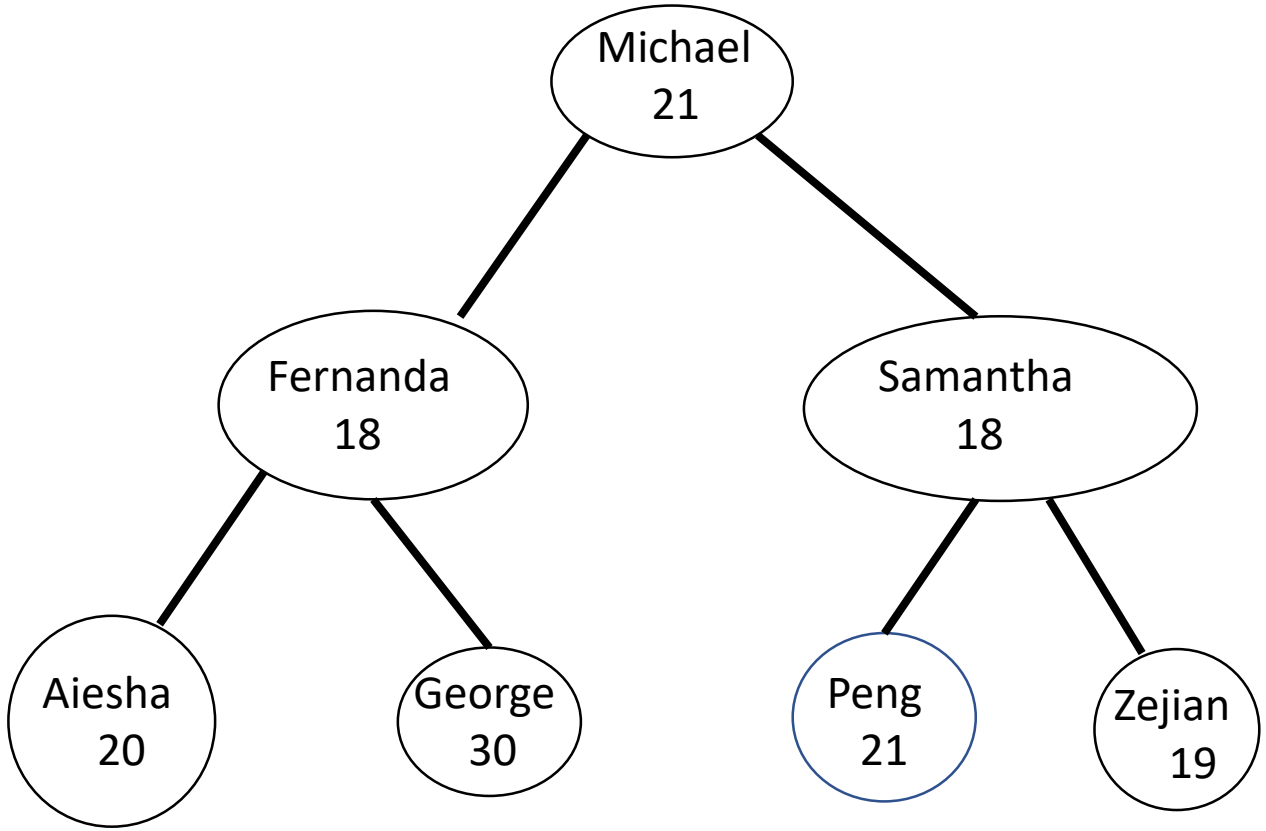
Michael 21	Fernanda 18	Samantha 18	Aiesha 30	George 30	Peng 21	Zejian 19

To find the value associated with a given key in this structure with n key-value pairs we would do a linear search looking for our key; this runs in time O(n).

If we sort the entries according to the key value we can do binary search rather than linear search; this reduces our lookup time to O(log(n)). Unfortunately, keeping the keys in sorted order when we add data makes the insertion method O(n).

Java has two better solutions.

A TreeMap stores the keys and their associated data in a *Balanced Binary Search Tree*. This guarantees both lookup and insertion times of O(log(n))



A HashMap goes back to the idea of storing the keys and data in an array, but this time there is a function called the *hash function* that helps you find the entry for any given key quickly. If you are able to store the map in a sufficiently large array, insert and lookup are *on average* both O(1).

Michael 21	Fernanda 18	Samantha 18	Aiesha 30	George 30	Peng 21	Zejian 19

Java implements these as TreeMap<K, V> and HashMap<K, V> where K is the class of the keys and V is the class of the associated values. The values can come from any class. For a TreeMap the key class K must implement the Comparable interface so keys can be compared. HashMaps can use any class for K.

TreeMaps and HashMaps have the same methods and may be used interchangeably, but each has situations where it is the preferable map. We will look at the details of both; try to get a sense of when it is best to uses each kind of map.