Discussion Questions for “Using Collections”

1. Now that you understand a little about how hash maps work, why does a HashSet use a HashMap as it’s backing store? What would you use as your HashMap key? What would you use as your HashMap value?

2. The professor has talked a little about “log(n) time” or “constant time” when describing the performance of some of the Collection concrete implementations. What do these terms mean?

3. The backing stores described in the Using Collections lecture include bit vectors, hash maps, tree maps, arrays, linked lists, and vectors. Is it important to understand all the details about how each of these backing stores are implemented? Is it good enough to know the basic idea, and the advantages and disadvantages of these constructs?

4. Can you think of some other kinds of backing stores that might be used to implement concrete implementations of some of the Collections interfaces? If so, what kind of backing store would you use, and which Collections interface would you use it for? What would be the advantages of your backing store? What about the disadvantages or restrictions?

5. A hash table is one of the super-algorithms in the computer science world because it allows you to quickly add and look up data in a data structure with a reasonable memory footprint, and an arbitrary key definition. Hash maps are also known as dictionaries, or content addressable memory, and can be found in almost all modern languages. In fact, the concept is so built in to Java that every object has built-in haschCode and equals method support using the Object class. Is it worth all the effort on the part of Java and Java coders to make sure than every object can be used as a hash map key? Discuss.