Week Five Reading Notes

Searching and Sorting

Week 5 is about searching and sorting. This is a classic computer science topic. Computer scientists have spent many years studying search and sorting algorithms and it is safe to say that there have been thousands of articles published on the topic.

It is often useful to think about the algorithms we use in life with physical objects in order to get an understanding of the algorithm to use for a computer program. For me, the term searching brings to mind the all too common morning occurrence where I rush around the house searching for my keys. I eventually find them (in the last place I look), but I wish it did not take so much effort. On weekends, you may find me sorting the laundry into separate piles of light and dark clothes, dreaming of a robot that can do this task. Although these are examples of searching and sorting, I certainly do not go about these tasks in any systematic way.

So let's consider an example that is closer to a task a computer might perform. Suppose I have a deck of playing cards. Recall that there are 52 cards in a deck.

I want to make sure the Queen of Hearts is in the deck, but the cards in the deck are scrambled (they are shuffled – ie, not sorted). To find the Queen of Hearts, I will have to search each card until I either find the Queen of Hearts or have looked through all the cards. On the average, it takes about 26 tries (52/2), but in the worse case, one must look at 52 cards.

This is a **linear time algorithm**. The search time is proportional to the number of cards. There is a famous series of children's books called "Where's Waldo" and each page contains a complex scene of many people and objects. Again a systematic search requires one to look at each object. Many people, however, never find Waldo. It is worse than linear time. What search strategy do you think they use?

There are many ways to sort the cards such that all the Ace's come first, ordered as hearts, diamonds, clubs, and finally the spades. Then come the two's, again in the same order. Followed by the three's and so on. Think about the different ways you might sort the deck to result in this ordering. There are many different ways, but some are more efficient than others.

As you follow the lecture, try to relate the algorithms to the playing card example. I bet some of you will come up with methods that do not correspond to those covered in Lecture. I would like to see these discussed in the forum.

Searching is a very common computer action. Google, Bing, Amazon, Netflix, and nearly every other web site has some type of search function – and they want the search to be fast! Sure, Python provides a "Find" function as well as a "Sort" one. When I interview programmers, I sometimes ask them to tell me how to sort a list of customer names. One recent candidate answered as follows:

```
Customers.sort()
```

You can decide if this person was offered the job.

The searching and sorting problem is important because the problem is easy to state. It is also easy to state the assumptions. Best of all, the algorithms themselves are not very complex! I think they are clever and fun. The techniques of searching and sorting are used over and over again in many different settings.

Finally, note that Search and Sorting are covered in Chapter 10 of the textbook.

- Larry Rudolph