More algorithms

Principles of Computing (15-110)

Instructor: Giselle Reis

Lecture on Tuesday 14th January, 2020

For all the problems below, think about the solution before reading it. Understanding a solution is different than being able to come up with it. And in this course, you will be evaluated mostly on your ability to come up with solutions.

1 Searching

Suppose you are holding a set of cards in your hand. You know for a fact that they are numbered 1 to 100, but they are not sorted, and your task is to find the highest card in this set. How do you proceed?

If all the cards are there, then you know that 100 is the highest card, so we just look for it. The instructions for that could be something like:

1. Check if the first card is 100:
   (a) If yes: finish. The card is found.
   (b) If no: remove that card. Go back to step 1.

This is not a very insightful problem though... Let's try something harder. Imagine that there are 100 cards, numbered from 1 to 100, but you are holding only 80 of them (any 80 cards, not neessarily the ones numbered from 1 to 80). You are still tasked to find the highest card. How do you proceed?

I am sure you can accomplish this task, but if you were to write down the step by step instructions, what do they look like? One way of figuring this out is to perform the task again, very slowly, and asking yourself each time: why did I make this decision. After some thinking, you may have come up with something similar to the following:

1. Take the first card. This is the highest card so far.
2. Take the next card.
3. Check if it is greater than the highest card so far.
   (a) If yes: take the current card as the highest card so far. Go back to step 2.
   (b) If no: discard the current card and go back to step 2.
4. When there are no more cards, the highest card so far is the highest card in the deck.

Can you come up with the instructions for finding the lowest card?

2 Sorting

You have the same set of cards, but now you need to put them in order. If you have all the cards, you know exactly what is going to be the first card, then the second, then the third, and so on and so forth. In this case, you can find the card with number 1, and place it on a new pile, then the card with number 2, etc. Our step by step instruction could be something like:

1. Check if the first card is 1:
   (a) If yes: Put the card at the end of the new pile.
   (b) If no: Place the card at the end of the deck. Go back to step 1.
2. Check if the first card is 2:
   (a) If yes: Put the card at the end of the new pile.
   (b) If no: Place the card at the end of the deck. Go back to step 2.
3. Check if the first card is 3:
   (a) If yes: Put the card at the end of the new pile.
   (b) If no: Place the card at the end of the deck. Go back to step 3.

   ...

You get the point... This is going to be a very long (and boring) sequence of instructions. There are a lot of repeated steps, and the only thing that changes is the number we are looking for. We know that we need to repeat the steps for all numbers from 1 to 100, so this can be abbreviated as:

1. For all numbers from 1 to 100, repeat the following steps:
   (a) Check if the first card has that number:
      i. If yes: Put the card at the end of the new pile.
      ii. If no: Place the card at the end of the deck.

As a bonus, you have also instructions on how to sort the numbers in reverse order! You simply need to change step 1 to "numbers from 100 to 1".

What if not all the cards are there? If you don't know the numbers, it is impossible to tell what is going to be the first, second, third, etc. card. How would you sort them?

One typical way people sort cards is by having two piles: one sorted and another unsorted. All cards start on the unsorted pile, and the sorted pile is empty. Then, one takes the first card and place it on the sorted pile (because a pile with one card is naturally sorted). Next, one takes the second card, and check if it is lower or greater then the card in the sorted pile, placing it accordingly. This procedure continues until all cards are in the sorted pile.

If we try to write the sequence of instructions we followed automatically, it might end up something like:

1. Place all cards in an unsorted pile.
2. Designate a place for the sorted pile, which is currently empty.

3. As long as there are cards in the unsorted pile, repeat:
   (a) Take a card from the unsorted pile.
   (b) Find its place in the sorted pile.
   (c) Go back to step 3.

Step 3b is still a bit vague. How do we find the place of a card in a sorted pile of cards? We flip through the cards until finding one that is greater than the one we have at hand! Here are the instructions to do that:

1. Take a card to place in the sorted pile
2. Check if the first card is greater than the card you want to place
   (a) If yes:
      i. Place the card at the top of the pile.
      ii. Flip the temporary pile up and place it back on top.
   (b) If no:
      i. Take the first card and place it faced down on a temporary pile.
      ii. Go back to step 2.

The full sequence of instructions consists on the first part, with step 3b replaced by the second part.

If you have the steps to find the lowest card, as suggested before, you can also use that to sort them. Simply take the lowest card from the unsorted pile, and place it on the sorted pile, faced down. Try writing down the steps for that.

3 Merging

Let's say that you are lazy and have two helpful friends. You decide to give each of them half of the cards to sort, hoping to make the task easier for you. After they finish, you have two sorted pile of cards. How do you merge them such that the final pile is sorted?

You know that the first card on each pile is the lowest card of that set of cards. Then we can conclude that the lowest card of the two will be the lowest card overall, right? (Convince yourself that this is true.) We can take that card out as the first one. Which card do we choose next? At this point we are basically left with the same problem as before: what is the lowest card overall among the cards in the two sorted piles? That means we can use the same reasoning!

In terms of sequences of steps, we could write something like this:

1. Check if both piles still have cards:
   (a) If yes:
      i. Take the lowest card among the two top ones.
      ii. Place it at the end of the final pile.
      iii. Go back to step 1.
   (b) If no: check if one of the piles still have cards

i. If yes: take the cards from the remaining pile and place them at the end of the final pile.

ii. If no: we are done.

What if you have more than two helpful friends. Suppose you have 8 of them. How can you minimize the amount of work for everyone? (Your friends will also be relieved, since each of them will have fewer cards to sort.) You can split the cards into 8 sets (it is not important if they have different sizes), give it to your 8 friends. Then you have 8 piles of sorted cards. How to merge all 8? Merging all of them at once will be quite intricate... Instead, it is easier if you merge them 2 by 2. There will be 4 pairs of sorted cards to merge. Well, you can ask for your friends' help again, now recruiting 4 of them. At the end, there will be 4 piles of sorted cards. How to merge all 4? You can use the same strategy again!

Many times, algorithmic solutions for problems will involve repeating the same step over and over again. Learn how to find these patterns and repetitions, so you can write your sequence of instructions (and in the future, your code) more concisely. Remember that this is a process. You can start with a very silly and repetitive solution, and refine it as much as needed (without breaking it, of course).

