Week 7b Searching and Sorting

Part 1 Sorting

Given a list `lst` of `n` numbers and an index `i` where `0 < i < n`, we can compare the two numbers `lst[i-1]` and `lst[i]`. If the two numbers are different, the bigger one should be swapped to the right such that `lst[i] > lst[i-1]` after the swap.

Write a function `bubble(lst)` which uses a simple loop to perform the above task from `i = 1` to `i = n-1`. (Note that the number of iterations is `n-1` and not `n`.)

```
>>> L = [4,5,6,7,1,2,3,9,8]
>>> L1 = bubble(L)
>>> L1
[4, 5, 6, 1, 2, 3, 7, 8, 9]
>>> L2 = bubble(L1)
>>> L2
[4, 5, 1, 2, 3, 6, 7, 8, 9]
>>> L3 = bubble(L2)
>>> L3
[4, 1, 2, 3, 5, 6, 7, 8, 9]
```

Did you notice something? What happens if `bubble()` is applied to a list many times?

How many times do I need to apply `bubble()` to a list in order to make the list fully sorted?

Write a function `bubbleSort(lst)` which returns a list that is sorted from the elements of `lst`. Here is some sample output in which you should be able to change `n` to a larger number and the sorting will still work.

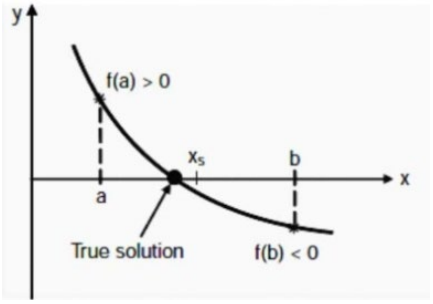
```
>>> from random import randint
>>> n = 20
>>> L = [randint(0,10000) for i in range(n)]
>>> print(L)
[8753, 4935, 9379, 7034, 515, 854, 7747, 3661, 9932, 1590, 8123, 3924, 9565, 469
9, 6735, 1109, 9955, 1600, 2481, 9363]
>>> print(bubbleSort(L))
[515, 854, 1109, 1590, 1600, 2481, 3661, 3924, 4699, 4935, 6735, 7034, 7747, 812
3, 8753, 9363, 9379, 9565, 9932, 9955]
```

Final Thoughts

Do you really need to…

- …apply `bubble()` so many times…
- …and to the entire list?

Part 2 Searching



Note: you cannot use 'Newton's method' as illustrated in the lecture slides. You will have to apply the bisection method for this part.

The ***bisection method*** in mathematics is a root-finding method that repeatedly bisects an interval and selects a subinterval in which a root must lie for further processing. Given a function $f(x)$, we want to solve for x when $f(x) = 0$.

In this question, we assume that the function f is continuous. Given two numbers a and b such that $f(a) > 0$ and $f(b) < 0$, we repeat the following:

- Compute $x_s = (a + b)/2$
- If $f(x_s) < 0$, then the solution lies on the left of x_s. So, $b = x_s$.
    - Otherwise, $a = x_s$.

We stop the above when the absolute value of $f(x_s)$ is smaller than a constant "error". After we exit the loop, the value of x_s should be very close to the actual solution of $f(x) = 0$. If $f(a) < 0$ and $f(b) > 0$, we just need to reverse the above inequality. If both $f(a)$ and $f(b)$ have the same sign, we can just "give up" as there might not be a solution which means the bisection method might run in an infinite loop.

You are given two functions f and g:

```
def g(x):
    return (x/40)**5 - 8*(x/40)**3 + x/4 + 6

f = lambda x : x**3  - 10*x + 4
```

Write a function `bisection(start,end)` to solve for x in both $f(x) = 0$ and $g(x) = 0$ with the bisection method. (In general, you should be able to solve any continuous function with $f($`start`$)$ and $f($`end`$)$ having different signs.) You can set "error" as a small number e.g. 0.00000001. The roots of $f(x) = 0$ and $g(x) = 0$ are -3.3459632955491543 and 63.39649252593517 respectively. (Your numbers could be a little off, but `abs(f(x))` should be less than "error".)

Part 3 Advanced Sorting (Extra)

This part is totally extra (for this course), but it would be interesting to try different sorting methods.

We are given a list `lst` of `n` numbers. To simplify the problem, we assume every number in the list is unique. (However, it is not difficult to solve the problem even if there are duplicate numbers in the list.)

We pick any number from `lst`, say `x`. For the rest of the numbers, we then separate them into two lists: `lsta` which contains all the numbers smaller than `x`, and `lstb` which contains the rest of the numbers. Let's give a name to this functionality: partition.

Then we apply some "magic" to `lsta` and `lstb` such that they become sorted after the "magic". Finally, we output the list `lsta + [x] + lstb`.

What is this "magic"? And what is the magic that we have performed?