Sorting

Introduction to Computer Programming

Dr. Paul Vrbik

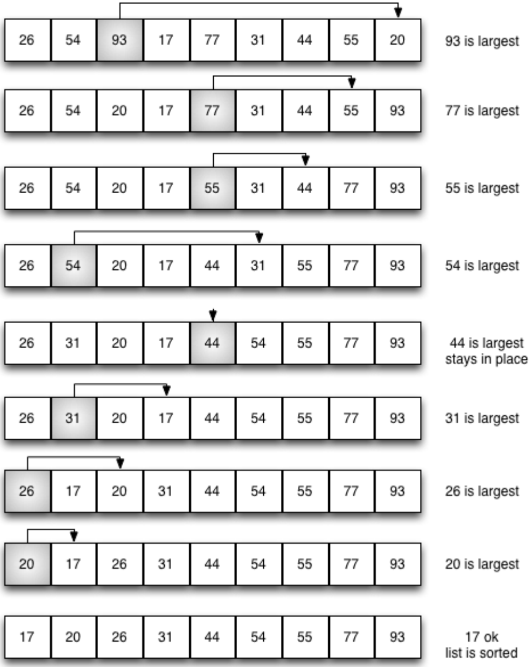
November 16, 2018

Question

Given a list of integers `xs` how can we order this list in ascending (equivalently descending) order?

Answer

Find (i.e. select) the largest member of `xs` and make it the last element. Then find the next largest and make it the second last etc.



26	54	93	17	77	31	44	55	20	93 is largest
26	54	20	17	77	31	44	55	93	77 is largest
26	54	20	17	55	31	44	77	93	55 is largest
26	54	20	17	44	31	55	77	93	54 is largest
26	31	20	17	44	54	55	77	93	44 is largest stays in place
26	31	20	17	44	54	55	77	93	31 is largest
26	17	20	31	44	54	55	77	93	26 is largest
20	17	26	31	44	54	55	77	93	20 is largest
17	20	26	31	44	54	55	77	93	17 ok list is sorted

Tasks

1. Implement selection sort in two meaningfully different ways.
2. Write a function that times how long (in seconds) another function requires to return when given input.
3. Experiment timing with lists of various lengths. Is one of our ways faster?

timeit.timeit

```
>>> from timeit import timeit

>>> def foo(n:int):
...   return 3**10**n

>>> timeit(lambda:foo(6), number=3)
0.21584114399999998
```

This is machine dependent.

```
>>> timeit(lambda:foo(6), number=3)/3
0.071947048
```

Average call-time for one call.

Note that `timeit` can only time functions with no arguments. Do `lambda:foo(6)` to test `foo` on some specific input.

Tasks (If Time)

1. Count the number of comparisons done in the selection sort algorithm.

Next Time

1. Bubble and Insertion sort.

2. Algorithm Complexity.