

Generative Recursion

Programming Language [ISL+](#)

Purpose To use generative recursion to solve interesting problems and to translate an algorithm into code.

Setup Remember that when a function doesn't follow structural recursion (i.e. the template) special care must be taken to ensure the function terminates.

Exercises

Exercise 1 Very often statisticians want to know what the n th smallest element in a list of numbers (such as quartiles or medians). A naive approach to this problem would sort the entire list in ascending order and get the n th element. However, this does much more work than necessary.

Design the function `nth-smallest`, which takes in n and a non-empty list of real numbers, where n is assumed to be less than the length of the given list. It works as follows:

1. If there is only one element in the list, return it
2. Partition the list into sublists of size 5 (with one possibly smaller list at the end)
3. Get the **exact** median of each of these sublists by individually sorting them and then selecting the middle element (as these sublists are all very small, this won't be a large computational cost)
4. Get the median of these medians using `nth-smallest`; this is now the pivot element
5. Partition the initial list into everything less than or equal to the pivot (besides the pivot itself), the pivot, and everything bigger than the pivot
6. Based on the number of elements that are less than or equal to the pivot and n , either return the pivot element or make the appropriate recursive call to `nth-smallest`

Note: to clarify, if n is 0, you should return the smallest element in the list.