Lecture 13: Analysis
10:00 AM, Oct 3, 2018

Contents

1 Putting things in order ........................................... 1

2 Induction .................................................................. 1

3 More Order of Growth ........................................... 1

4 Insertion sort .......................................................... 1

5 Quiz ........................................................................ 2
   5.0.1 Solution ........................................................ 2

6 Summary ................................................................... 2

1 Putting things in order

input: a list of numbers
output: a list with the same elements in ascending (sorted) order

Today we’re going to examine two simple strategies to sort things and then a more complicated one.

2 Induction

See the slides and lecture capture for a detailed explanation of the process of proving by induction.

3 More Order of Growth

Let $g(n)$ be a function. We say that another function $f(n)$ is $O(g(n))$ if there exist nonnegative constants $n_0$ and $c$ such that, for every value of $n$ that is greater than $n_0$, $f(n)$ is less than or equal to $c \cdot g(n)$.

For example, suppose $g$ is the function that on input $n$ returns as output $n$, i.e. we write $g(n) = n$. In this case, to say that $f(n)$ is $O(g(n))$ means that there are constants $n_0$ and $c$ such that $f(n) \leq c \cdot n$. I’ll prove that this is the same as saying that a procedure with running-time function $f(n)$ is a linear-time function.
4 Insertion sort

The idea behind insertion-sort:

1. Remove the car of the input list and recursively sort the cdr.
2. Insert the car into the right place in the sorted list.

Example:

input: (2 1 9 6) 
   recursive input: (1 9 6) 
   recursive output: (1 6 9) 
   output: (1 2 6 9)

It is easy to derive the recursive input from this...
...take the cdr! (O(1))
This is given according to the spec.
insert-in-order
It is harder to derive this output (O(n))

5 Quiz

Write insert-in-order

input: number x, list lst of numbers in ascending order
output: list consisting of x and elements of lst, all in ascending order.

5.0.1 Solution

(define insert-in-order
  (lambda (x lst)
      ....... ))

6 Summary

- We can use induction to prove the worst case run times of functions.
- The insertion-sort algorithm has a runtime of $O(n^2)$, where $n$ is the length of the list and is used to sort a list of integers.