CPSC 320 Worked Example, Asymptotic Analysis

September 30, 2016

1 Intro

You should have done a substantial amount of asymptotic analysis of algorithms and code prior to CPSC 320, but you'll want to review. To get you started, we'll work through analysis of a very messy algorithm below.

This example is **NOT typical** in several ways: **It's not doing anything useful.** In part because of this, the algorithm is absurdly complex in its structure and ridiculous simple in the actual operations it performs. We'll usually analyse useful algorithms! **Our analysis is not with respect to the size of the input** but with respect to the actual value of the input. (I.e., rather than having as input something interesting that is of size n like a graph with n nodes, this has as input the actual number n.) . **We don't even consider analysing anything but worst-case runtime.** This **is** the most common thing we'll analyse, but we'll sometimes be interested in other properties like average- or expected-case runtime, worst-case memory usage, or worst-case number of proposals made (in stable marriage!).

Take this strange "algorithm" as an opportunity to think about steps you might take in asymptotic analysis.

2 Count-A-Bunch

Analyse the worst-case asymptotic runtime of `CountABunch` in terms of n.

```
// ASSUME: n is an integer and n > 0
CountABunch(n):

  i = 1

  while i < n:

    i *= 2

  i /= 2

  while n is not divisible by i:

    i++

  for j = 1 to i:

    for k = 1 to i:

      print(DescendFrom(j, k))


DescendFrom(n, x):

  if n == 0:

    return x

  else:

    return DescendFrom(floor(n/2), x + 1)
```