TCSS 343 - Week 2 - Wednesday

Jake McKenzie

January 23, 2019

Asymptotics with an Introduction to the Analysis of Algorithms

"There is nothing radical about moral clarity".

...

Alexandria Ocasio-Cortez

"I am incapable of conceiving infinity, and yet I do not accept finity. I want this adventure that is the context of my life to go on without end".

...

Simone de Beauvoir

"Man has not given himself the taste for the infinite and the love of what is immortal. These sublime instincts do not arise from a caprice of the will; they have their unchanging foundation in his nature; they exist despite his efforts. He can hinder and deform them, but not destroy them."

...

Alexis de Tocqueville

0. Prove the theorem below using the techniques of binding the term and splitting the sum to find a tight bound for the sum. Make sure your proof is complete, concise, clear and precise.

Theorem 0:

$$\sum_{i=1}^{n} i^4 \in \Theta(n^5)$$

1. Prove the theorem below using the techniques of binding the term and splitting the sum to find a tight bound for the sum. Make sure your proof is complete, concise, clear and precise.
   Theorem 1:
   $$\sum_{i=1}^{n}(\log_2 i)^2 \in \Theta(n(\log_2 n)^2)$$

Here is an algorithm for singing that annoying song "99 Bottles of Beer on the Wall". It exhibits all the qualities we expect from algorithms, namely that it is explicit, precise, unambiguous, mechanically-executable sequence of elementary instructions intended to accomplish a specific purpose.

```
BottlesOfBeer(n):
  For i ← n down to 1
      Sing "i bottles of beer on the wall, i bottles of beer,"
      Sing "Take one down, pass it around, i − 1 bottles of beer on the wall."
  Sing "No bottles of beer on the wall, no bottles of beer,"
  Sing "Go to the store, buy some more, n bottles of beer on the wall."
```

2. If you use this algorithm to sing this terrible song, n-times how many times will you say beer? $2n$?

Here is an algorithm for singing a far less annoying song, 12 Days of Christmas, generalized to n Days of Christmas. Instead of analyzing the exact ~~running~~ singing time, let focus on the asymptotic behavior.

```
NDaysOfChristmas(gifts[2..n]):
    for i ← 1 to n
        Sing "On the ith day of Christmas, my true love gave to me"
        for j ← i down to 2
            Sing "j gifts[j]"
        if i > 1
            Sing "and"
        Sing "a partridge in a pear tree."
```

3. How many gifts are sent into this algorithm? Express this as an exact answer.

4. During the first n days of Christmas, my true love gate to me exactly how many gifts? Express this as a sum, then solve that sum finding the the big theta space complexity of the number of gifts my true love gave to me.

5. Write does and solve the recurrence for the running time of an algorithm I've run into in the wild.

$$T(n) = 3T(\frac{n}{4}) + n^2$$

6. Write does and solve the recurrence for the running time of the quick-select algorithm, used within quicksort.

$$T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + n$$