Algorithms

Part 3: Time Complexity Basics

Constant, Linear, and Quadratic Time

Notes for CSC 100 - The Beauty and Joy of Computing
The University of North Carolina at Greensboro

Reminders

Blown to Bits:
Chapter 3 on-line discussion contribute by Wednesday!

Homework:
Homework 2 due in one week: Wed., Oct. 1

On the horizon: Midterm Wednesday, Oct. 8

Constant time

We say a script (or part of a script or block definition) takes constant time if it is a constant (usually small) number of basic steps, regardless of input.

Question: Are all of these constant time?
What about loops?

The number repetitions depends on length of "values"

- So this is not constant time...

Constant time operations, repeated "length of input" times is **linear time**

Mathematically: Constant time loop body is time "c"
- Repeated "n" times where n is length of list
- Total time is then c*n (that's a linear function!)

General iterator pattern

On previous slide:

- Time was expressed as a function of input size
- Could write time as \( T(n) = c \cdot n \)

In general:

We know how many times it repeats, and all basic blocks are constant time except perhaps our "do something..." block

- In general, if time for "do something..." block is \( T(n) \), then time for complete script with loop is \( n \cdot T(n) \)
- If "do something" is constant time, total time is \( c \cdot n \) (linear)
- If "do something" is linear time, total time is \( c \cdot n^2 \) (quadratic)

Two challenges
Plotting the Running Times

Measured (using BYOB) and calculated running times for max pos:

Note: The straight line of this graph should remind you of linear function graphs from math class!

Plotting the Running Times

Measured (using BYOB) and calculated running times for sort:

Note: The nice smooth parabola should be familiar to you from math class!

Another challenge

The following predicate tests whether a list has any duplicates:

Question: What's the time complexity?
Predicting Program Times - Linear

Basic idea: Given time complexity and sample time(s) can estimate time on larger inputs

Linear time: When input size doubles, time doubles
When input size triples, time triples
When input size goes up by a factor of 10, so does time

Example: A linear time algorithm runs in 10 sec on input size 10,000
How long to run on input size 1,000,000?

Answer: 

\[ \frac{1,000,000}{10,000} = 100 \text{ times larger input} \]

Therefore 100 times larger time, or \( 10 \times 100 = 1,000 \) sec
Or \( \frac{1,000}{60} = 16.667 \) minutes

Predicting Program Times - Quadratic

Basic idea: Given time complexity and sample time(s) can estimate time on larger inputs

Quadratic time: When input size doubles (2x), time quadruples (4x)
Input size goes up by a factor of 10, time goes up \( 10^2 = 100 \) times
Input size goes up \( k \) times, time goes up \( k^2 \) times

Example: A quadratic time algorithm runs in 10 sec on input size 10,000
How long to run on input size 1,000,000?

Answer: 

\[ \frac{1,000,000}{10,000} = 100 \text{ times larger input} \]

Therefore \( 100^2 = 10,000 \) times larger time, or 100,000 sec
Or \( \frac{100,000}{60} = 1666.7 \) minutes (or 27.8 hours)

Predicting Program Times - Your Turn

Joe and Mary have created programs to analyze crime statistics, where the input is some data on each resident of a town

- Joe’s algorithm is quadratic time
- Mary’s algorithm is linear time
- Both algorithms take about 1 minute for a town of size 1000

Both would like to sell their program to the City of Greensboro (population 275,000)

Problem: Estimate how long each program would take to run for Greensboro
Summary

- Algorithm "time complexity" is in basic steps
- Common complexities from this lecture, from fastest to slowest are constant, linear, and quadratic
  - A single step, or sequence of constant-time blocks is constant time
  - A simple loop with constant time operations repeated is linear time
  - A loop containing a linear time loop is quadratic
- Speed depends on algorithm time complexity
  - Constant time is great, but not many interesting things are constant time
  - Linear time is very good
  - Quadratic time is OK
- Given time complexity and one actual time, can estimate time for larger inputs