CMPT307: Introduction

Week 1-2

Xian Qiu

Simon Fraser University

xianq@sfu.ca

Organization

Data Structures and Algorithms

Lectures

- 36 lecture hours: 3 hours per week (last lecture on Dec 5, 2016)
- online material: `https://courses.cs.sfu.ca/2016fa-cmpt-307-d3/pages/`

Your assignments

- following the lectures
- reading the literature
- solving exercises

Organization

Grading

- homework 20%
- three quizzes 30% (50 minutes each)
- final exam 50% (3 hours)

Textbook

Introduction to Algorithms - 3rd Edition, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, MIT Press, 2009.

Data structures visualization:

`https://www.cs.usfca.edu/~galles/visualization/`

About Homework

- do not copy answers from any source
- done in teams of two students
- weekly release
- submit within two weeks
- short and precise solutions

Office Hours

Xian Qiu

- TASC 1, room 9025
- office hour: Monday 16:30 - 17:30 or by appointment

TA

- Cong Zhang
- email: congz@sfu.ca
- room: CSIL, ASB 9838.1
- office hour: Friday 16:30-17:30

Overview

- lists, queues, stacks
- dictionaries: binary search trees, red-black trees
- sorting: heapsort, quicksort, non-comparison sort etc.
- priority queues: heaps
- randomized algo., average case analysis, amortized analysis
- dynamic programming
- greedy algorithms
- graph algorithms
- NP-completeness, approximation algorithms

Algorithms

- data structures
- what is an algorithm?
- what machine do we have?
- design and analyze machine independent algorithms

Model of Computation

RAM: random access machine

- single processor
- each simple operation (+, *, -, =, if, call) takes exactly one unit of time
- loops and subroutines are not simple operations
- each memory access is free
- unlimited memory

Problem and Instance

Example: sorting

given n integers, sort them in monotonically non-decreasing order

- a sorting instance: $7, 4, 3, 9, 8, 0$
- problem $=$ set of instances sharing the same "description"
- an algorithm solves a problem if its output is correct for all instances

Analysis of Algorithms

two ingredients

- efficiency: running time
- quality: quality of solutions

analyzing approaches

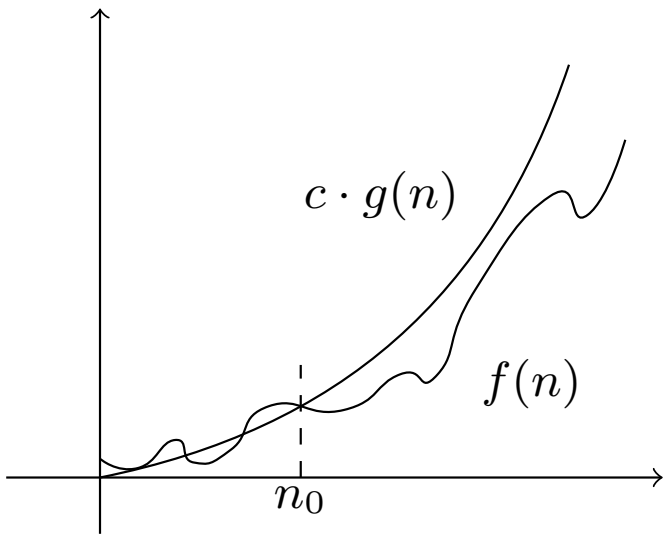
- empirical
  - computer, language and compiler dependent
- average case
  - distribution unknown, often hard
- **worst case**
  - performance guarantees, but too pessimistic
- etc...

Asymptotic Upper Bounds

Definition

$f(n) = O(g(n)) \Leftrightarrow \exists$ constant $c > 0$ and $n_0 \in \mathbb{N}$ such that
$f(n) \leq c \cdot g(n), \forall n \geq n_0$

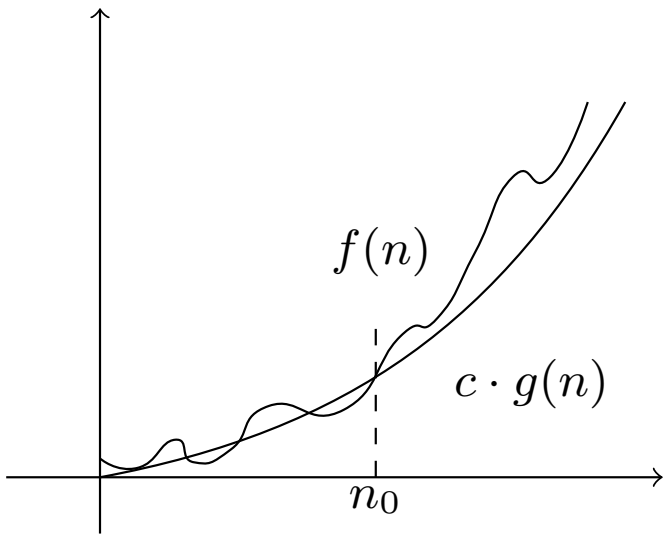
here "$=$" stands for "$\in$"



Asymptotic Lower Bounds

Definition

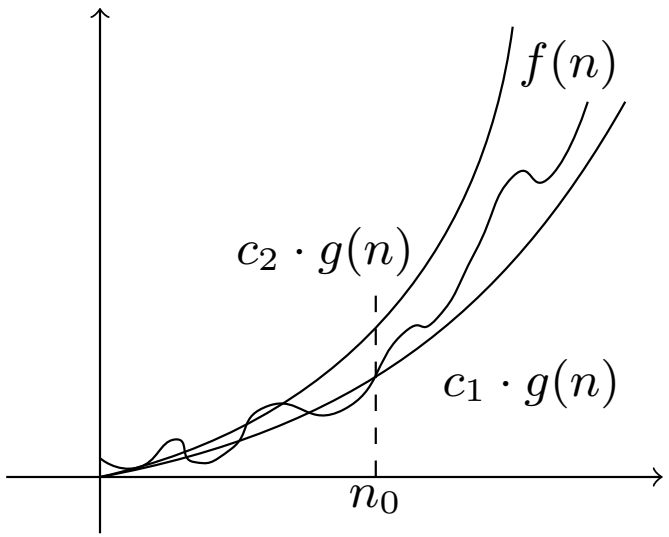
$f(n) = \Omega(g(n)) \Leftrightarrow \exists$ constant $c > 0$ and $n_0 \in \mathbb{N}$ such that
$f(n) \geq c \cdot g(n), \forall n \geq n_0$



Asymptotic Equivalence

Definition

$f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$



More Notations

Definition

- $f(n) = o(g(n)) \Leftrightarrow f(n) = O(g(n))$ and $f(n) \neq \Theta(g(n))$
- $f(n) = \omega(g(n)) \Leftrightarrow f(n) = \Omega(g(n))$ and $f(n) \neq \Theta(g(n))$

Examples

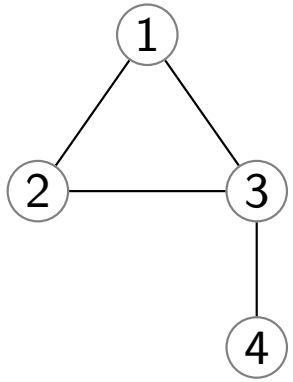
- $n \log n = O(n^2)$, $n \log n = o(n^2)$
- $2n^2 + 3n = \Theta(n^2)$
- $n! = \Omega(2^n)$, $n! = O(n^n)$

Stirling's formula

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

Why Data Structures Matter?

given a graph $G = (V, E)$, $|V| = n$, $|E| = m$



- adjacency matrix: $n^2 = 16$ bytes

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- adjacency list L: $n + 2m = 12$ bytes

$$L_1 \to \{2, 3\}$$
$$L_2 \to \{1, 3\}$$
$$L_3 \to \{1, 2, 4\}$$
$$L_4 \to \{3\}$$

Edge-problem

- ▷ given a graph $G = (V, E)$ and two vertices i, j
- ▷ question: does edge $\{i, j\}$ exist?

adjacency matrix A $O(1)$

if $a_{ij} = 1$, return yes; **else** return no

adjacency list L $O(n)$

for $k \in L_i$ **do**
  if $k = j$ return yes;
  else return no;