CS 170 Section 1
Asymptotic Analysis

Owen Jow | owenjow@berkeley.edu

Agenda

- Introduction
- Asymptotics review
- Divide-and-conquer

Introduction

About Me

- Owen Jow
- owenjow@berkeley.edu
  - but please only email me as a last resort, or if you have an administrative question
  - your alternatives for getting help include cs170@berkeley.edu, Piazza, OH, and your classmates
- Section: TBD
- OH: TBD

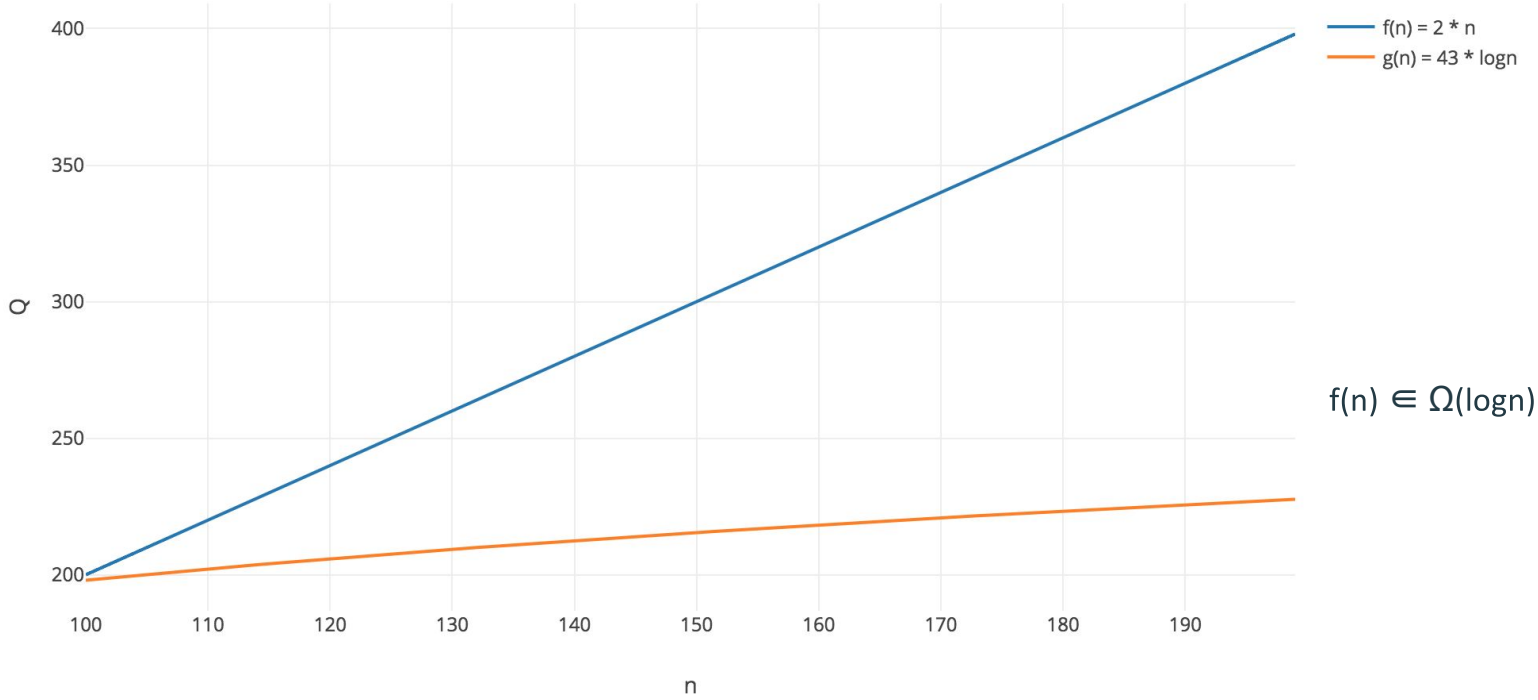
About CS 170

- Berkeley's premier CS algorithms course
  - Important for theory and software engineering interviews
- My main tip for success:
  - **More than anything else, focus on understanding and solving the homeworks on your own.** Start them early! *Immediately* after they come out, read all the problems and take the time to process them in your mind. Try to finish them as soon as possible, but hold off on asking other people for help. That being the case, you should be drawing out new approaches whenever you have time – it's okay to eat lunch with your problem set.
    - If you're keeping up with the homeworks, you're keeping up with the class.
    - If you're able to do the homeworks on your own, you're also in good stead for the exams!
- I also highly recommend the textbook (*Algorithms* by Dasgupta, PapaD, and our very own Vazirani).

Asymptotics Review

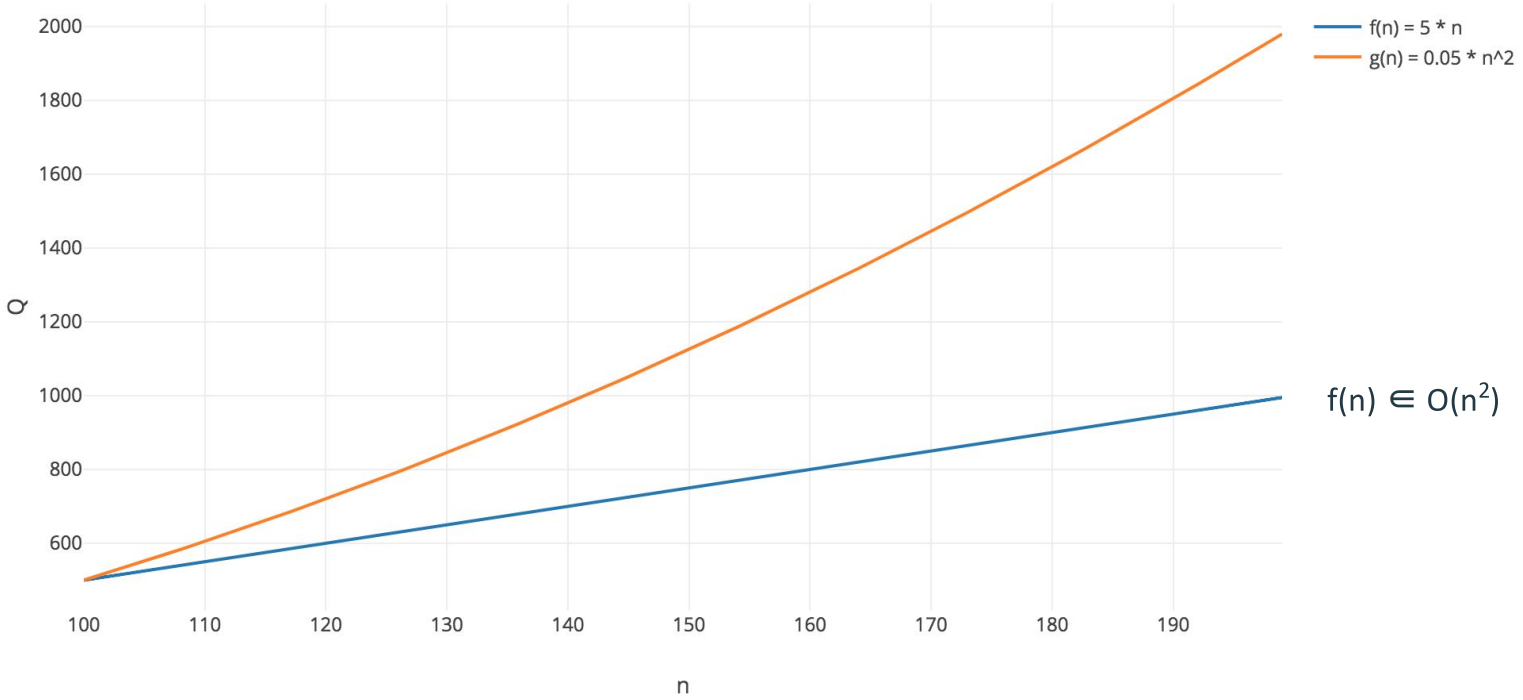
The Big-O Problem

- We are describing the **growth** of a quantity Q with respect to another quantity n
- We do this using functions: Q = f(n)
- However, it's messy to have every individual function be different ($5.9n^{3.1} + 66$, anyone?)…
  ...so we separate our growth functions into classes, and place each individual function into a class via big-Ω, big-Θ, or big-O notation.
- Typical classes are functions like 1, n, or n^2.



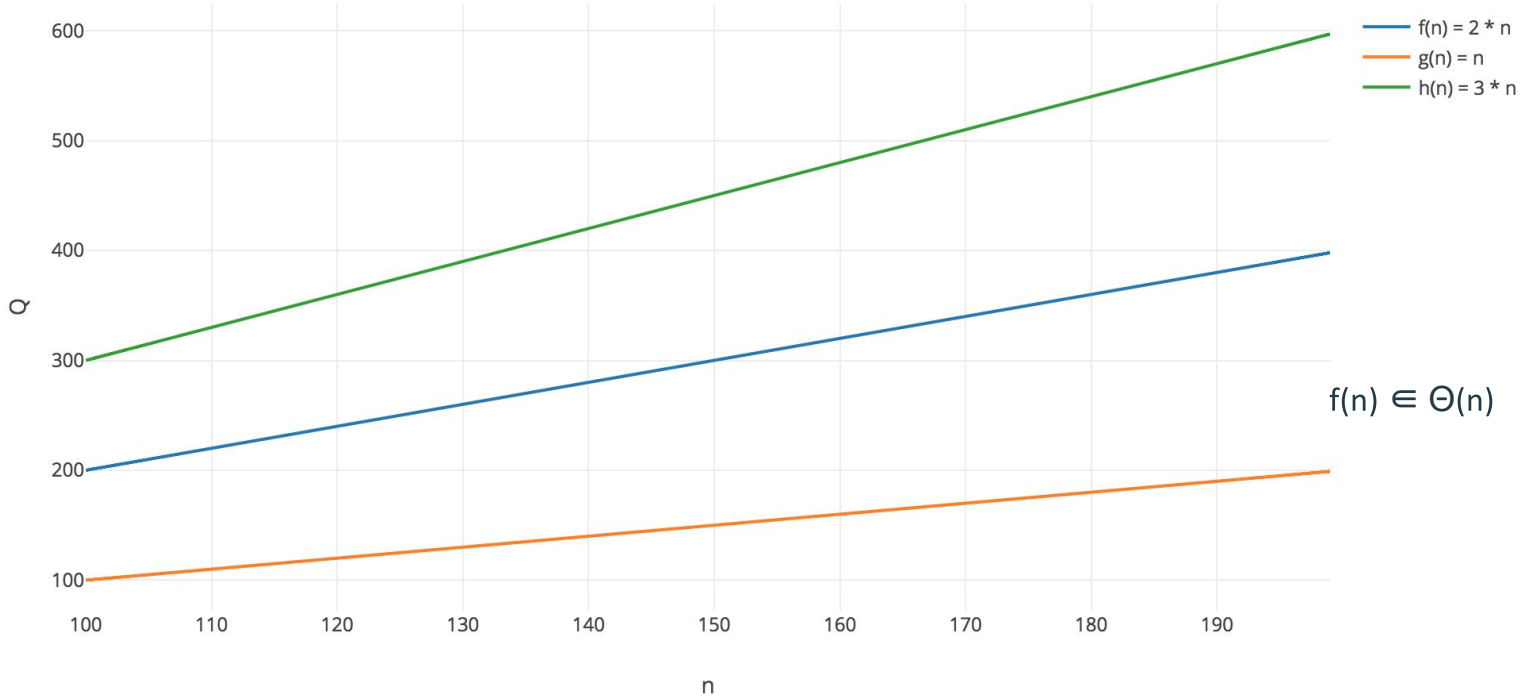
$f(n) \in \Omega(\log n)$

Ω is a lower bound.
It says “for large n, the true function **grows no slower** than a constant times this function.”



$f(n) \in O(n^2)$

O is an upper bound.
It says “for large n, the true function **grows no faster** than a constant times this function.”



$f(n) \in \Theta(n)$

Θ is both a lower bound and an upper bound.
It means both Ω and O at the same time.

Notes

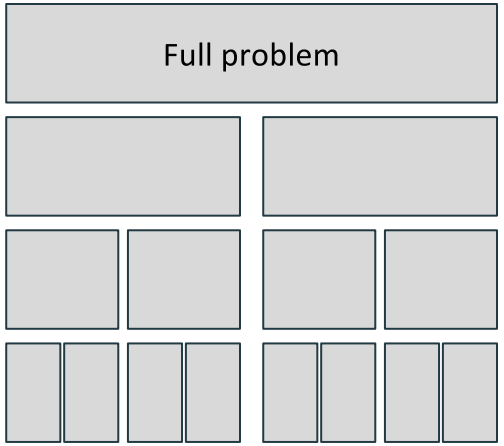
- In asymptotics problems, you can drop constant multipliers and addends.
  Or keep them in. It’s all the same.
- In 170, unlike in the 61 series, we often consider complexity at the bit level.

Divide-and-Conquer

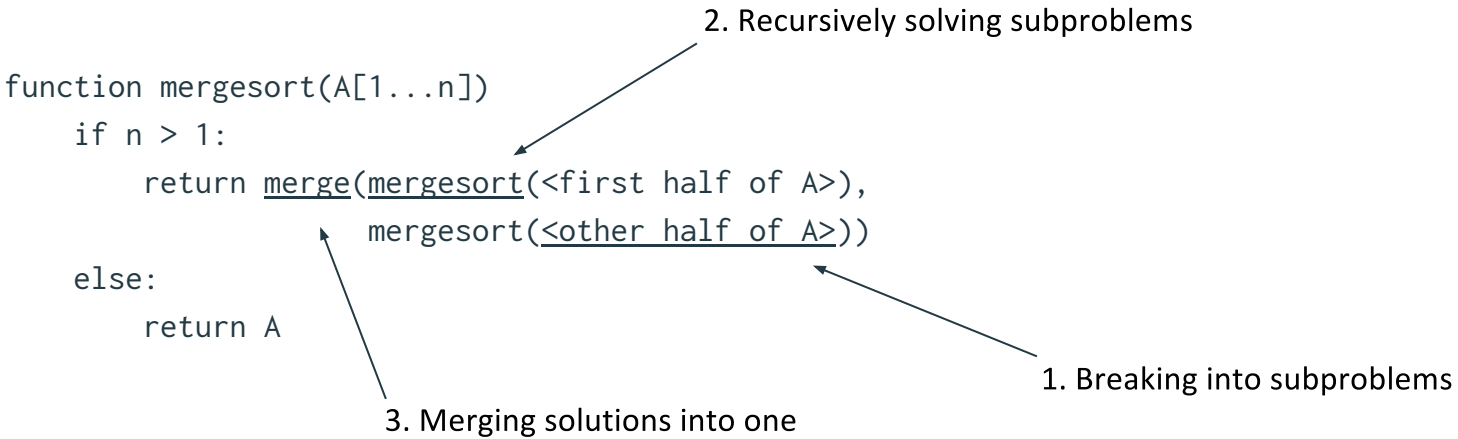
Divide-and-Conquer

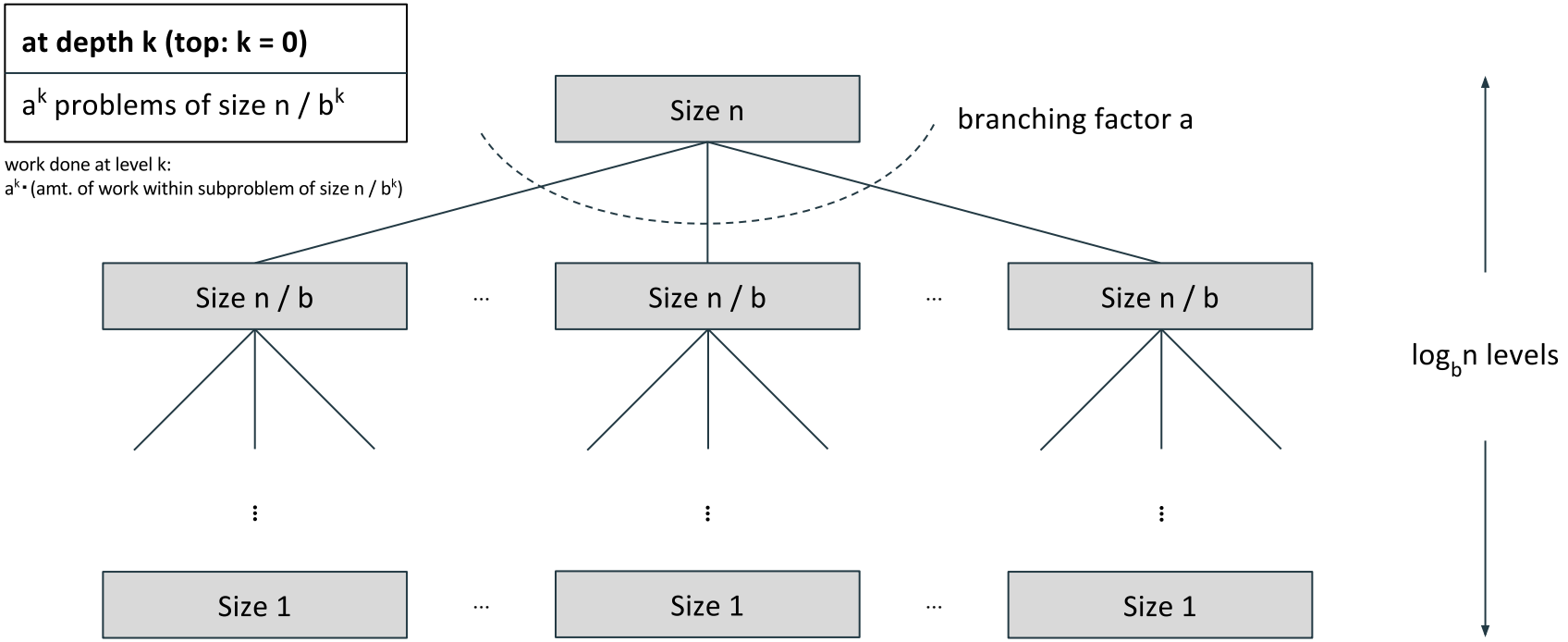
- A problem-solving approach that involves
  - **breaking the problem into smaller ones**,
  - **recursively solving the smaller problems**, and finally
  - **merging all of the solutions into one**.

Classic examples: binary search, merge sort



Merge Sort


2. Recursively solving subproblems
function mergesort(A[1...n])
if n > 1:
return merge(mergesort(<first half of A>),
mergesort(<other half of A>))
else:
return A
1. Breaking into subproblems
3. Merging solutions into one



With regard to asymptotic analysis for divide-and-conquer functions, I particularly enjoy the diagram and explanation from Chapter 2 of the textbook (reproduced above from *Algorithms* by Dasgupta et al.).