

Lab 8



Practical Asymptotic Analysis & TreeMaps

Announcements

- ▷ Lab 8 - due Friday 03/11
- ▷ HW 6 - due Tuesday 03/29
- ▷ Project 2: Ataxx released!
 - Checkpoint - due Friday 03/18
 - Project - due Friday 04/01

**Your health and well-being is
more important than this course**

Academic & Mental Health Resources

- ▶ [Official department academic resources](#)
- ▶ **Mental health resources**
 - [Counseling and Psychological Services \(CAPS\)](#)
- ▶ Contact your mentor TA if you just want to talk to someone
 - Keep in mind we are not trained counselors
- ▶ Life happens – we understand. Need an extension?
 - Request one through [Beacon](#)
 - Need additional support? Contact cs61b@berkeley.edu

CAPS Resources

- All students will continue to have access to mental health services at UHS **regardless of insurance** and UHS counseling visits are still free of charge.
- CAPS offers a wealth of resources:
 - Online self-help tools ([Therapy Assistance Online](#)) and wellness videos
 - [Let's Talk consultations](#)
 - Skill-based workshops
 - Support spaces for various communities
 - [Counseling Groups](#)
 - [Individual and couples counseling](#)
 - Referral to additional support resources on- and off-campus
- For students that want to see a counselor, the majority of appointments are now offered the same day.
- Students are seen for their top-of-mind concerns and given resources, whether at an initial or returning counseling visit.



Lab Overview

How do we measure the efficiency
of programs?

One method: algorithmic analysis

- ▷ Analyze programs from a theoretical math perspective
- ▷ You've seen this in lecture & discussion

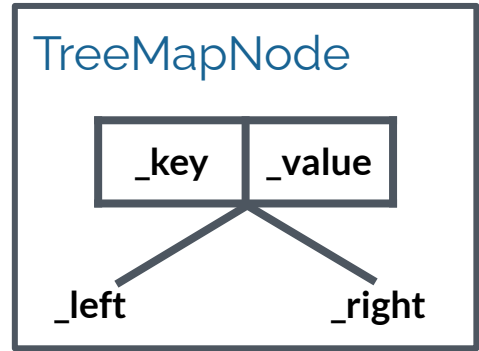
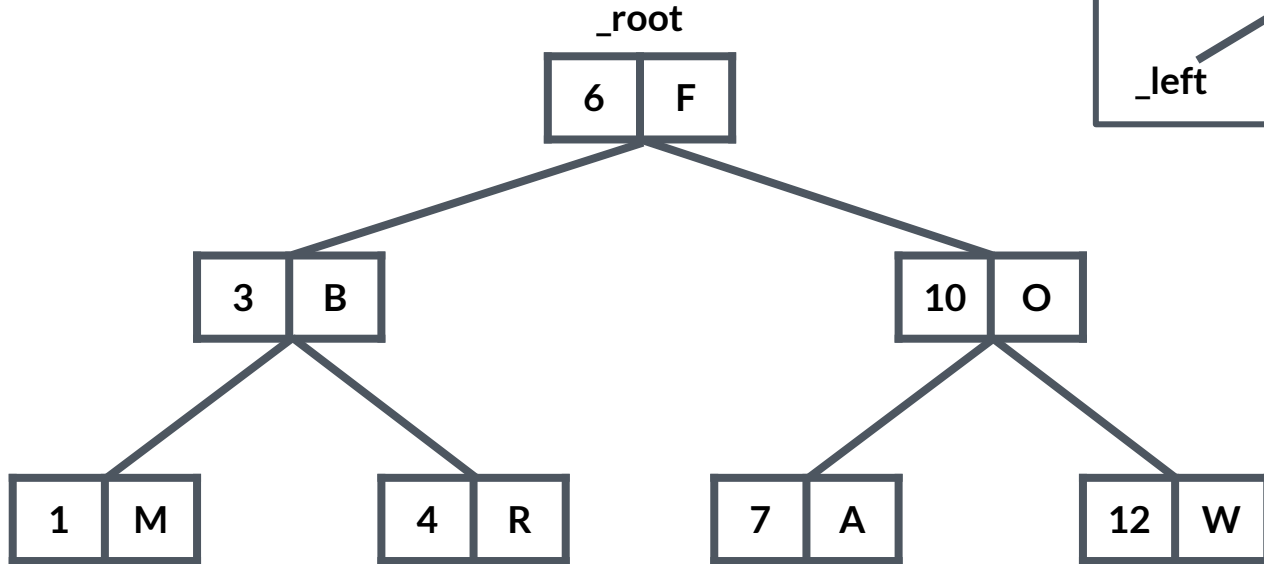
Another method: timing

- ▶ We can time how long it takes a program to run for different inputs
 - We provide `timing.Timer` to do the timing
- ▶ Gives a *very* rough estimate of runtime
 - Greatly affected by computer specs + other external factors
- ▶ For the lab, run `timer.SortTiming`
 - Play around with `N_TRIALS`, `BY`, `N_REPEATS`
 - Answer the questions in `sort_timing.txt`

Amortized Runtime

- ▷ In simplified terms: *on average*
- ▷ If unfamiliar: review [lecture 17](#)
- ▷ For the lab, run `timer.AmortizationTiming`
 - Play around with `MAX_SIZE`, `N_LISTS`, `ACCUMULATE`
 - Answer the questions in `amortization_timing.txt`

TreeMap



Sorted with respect to the keys, NOT the values!

compareTo

```
class A implements Comparable <A> {  
    @Override  
    public int compareTo(A other) { ... }  
}
```

```
A a1 = new A();  
A a2 = new A();  
  
if (a1.compareTo(a2) < 0) {  
    // a1 < a2  
} else if (a1.compareTo(a2) > 0) {  
    // a1 > a2  
} else if (a1.compareTo(a2) == 0) { // redundant if clause  
    // a1 = a2  
}
```