This thread is part of the proposed schedule for study for the first exam. Discuss the practice problems here.
View questions here: http://beastie.cs.ua.edu/concepts/cs/al/recurrences.html
This thread covers questions 1-35 of recurrences.

Work together on the proposed answers to questions on this shared Google Doc (comment reasoning/arguments behind answers)

Upcoming topics (threads start two days in advance): Recurrences 36-70, Feb 11 | SB trees 1-26, Feb 12 | SB trees 26-52, Feb 13 | Recurrences 71-105, Feb 14
Full schedule

---

Subject: Re: Concept Review: Recurrences (Part 1)
Posted by davidmccoy on Thu, 09 Feb 2017 19:46:25 GMT

Quote:1. Stooge sort has the following algorithm. Recursively sort the lower two-thirds of an array, then recursively sort the upper two-thirds, then recursively sort the lower two-thirds again. The recursion stops when the array consists of two or fewer elements. If the array size is two, the elements are swapped if necessary. Which of the following recurrence equations describe stooge sort?

There are 3 recursive calls, each call operates on 2/3 of the array, and there's no work done to combine the results of each recursive step. Help?

---

Subject: Re: Concept Review: Recurrences (Part 1)
Posted by davidmccoy on Thu, 09 Feb 2017 21:17:13 GMT

Quote:Number 11. Given the recurrence $T(n) = 4T(n/3) + \log(n)$, how much work is done at the bottom level?
So I thought this was how a recursion tree was drawn...

but then shouldn't the work done at the bottom level be (number of nodes) * $\log(n/3)$, which should be $4\log(\text{base 3})(n)$? Not sure what the answer is.
davidmccoy wrote on Thu, 09 February 2017 13:46

Quote: 1. Stooge sort has the following algorithm. Recursively sort the lower two-thirds of an array, then recursively sort the upper two-thirds, then recursively sort the lower two-thirds again. The recursion stops when the array consists of two or fewer elements. If the array size is two, the elements are swapped if necessary. Which of the following recurrence equations describe stooge sort?

There are 3 recursive calls, each call operates on 2/3 of the array, and there’s no work done to combine the results of each recursive step. Help?

Look closely at your recursive calls: 3T(3n/2). Does that make sense?

---

davidmccoy wrote on Thu, 09 February 2017 15:17

Quote: Number 11. Given the recurrence T(n) = 4T(n/3) + logn, how much work is done at the bottom level?

...but then shouldn't the work done at the bottom level be (number of nodes) * log(n/3), which should be 4log(base 3)(n)? Not sure what the answer is.

The work done by each node at the bottom level is usually theta(1).

---

davidmccoy wrote on Fri, 10 February 2017 21:06

I was putting 3/2 on the bottom of the fraction, but I see now that doesn't make sense. If I was operating on half the array, I would put 3T(n/2), where (1/2) is being multiplied by n. So the correct
I know this was discussed in class, but I still don't understand what exactly is meant by a master theorem falling "between" two cases. For example:

12) In terms of the master recurrence theorem, where does the equation \( T(n) = 4T(n/4) + n^2 \) fall?

It's almost case 3, but doesn't satisfy \( af(n/b) \leq cf(n) \). So would we say that it is between cases 2 and 3? If that's the criteria, how would we have an equation fall between cases 1 and 2?

---

Subject: Re: Concept Review: Recurrences (Part 1)
Posted by SSinischo on Sat, 18 Feb 2017 20:37:39 GMT
View Forum Message <> Reply to Message

cdyancey wrote on Sat, 18 February 2017 13:08 I know this was discussed in class, but I still don't understand what exactly is meant by a master theorem falling "between" two cases. For example:

12) In terms of the master recurrence theorem, where does the equation \( T(n) = 4T(n/4) + n^2 \) fall?

It's almost case 3, but doesn't satisfy \( af(n/b) \leq cf(n) \). So would we say that it is between cases 2 and 3? If that's the criteria, how would we have an equation fall between cases 1 and 2?

This is case 3. \( 4(n/4)^2 = 1*f(n) \)

---

Subject: Re: Concept Review: Recurrences (Part 1)
Posted by cdyancey on Sat, 18 Feb 2017 20:56:00 GMT
View Forum Message <> Reply to Message

^ c must be less than 1

---

Subject: Re: Concept Review: Recurrences (Part 1)
Posted by SSinischo on Sat, 18 Feb 2017 21:30:49 GMT
View Forum Message <> Reply to Message

On page 95 of the book, a similar recurrence is solved. You are right, it does fall in between case 2 and 3.

---

Subject: Re: Concept Review: Recurrences (Part 1)
Posted by lusth on Sat, 18 Feb 2017 23:05:50 GMT
View Forum Message <> Reply to Message

\[ T(n) = 4T(n/4) + n^2 \]

This is a solid case three (and the regularity condition holds).
Of course. Dumb mistake on my part.

Regardless of the poor choice in example, the big question remains: what exactly constitutes being "between" cases?

Whoops, I need to pay more attention when trying to answer questions! The recurrence equation on page 95 that I was talking about is an example of an in-between case:

\[ T(n) = 2T(n/2) + n\log(n) \]

I'm not sure if my reasoning is 100% sound, but here's how I'd approach it:

Log base 2 of 2 is 1, and \( n\log(n) \) is always going to be more than \( n^1 \), so we can't use case 2.

If we are to use case 3, consolidation/combination/\( f(n) \) dominates over recursion. We can see the relationship between the two by putting them into the ratio:

\[ \frac{f(n)}{n^{(\log base b of a)}} \]

In our question here, we have the ratio \( \frac{n\log(n)}{n^1} = \log(n) \). But is \( \log(n) \) asymptotically larger in-between.

Conversely, let's look at another equation in the book:

\[ T(n) = 3T(n/4) + n\log(n) \]

Log base 4 of 3 is about .792. The ratio \( \frac{n\log(n)}{n^{.792}} \) grows to infinity. Let's check if it's...
We know that \( n \log n > n^1 \), so as long as our denominator is less than that, this ratio will grow to

So that's how I determine in-between cases. Please correct me if something looks wrong here!

---

Subject: Re: Concept Review: Recurrences (Part 1)  
Posted by daweil on Mon, 20 Feb 2017 00:41:28 GMT  
View Forum Message <> Reply to Message

Can we get extra credit as a class for testing Linus's Law in a pseudo fashion with these shared documents of awnsers?

---

Subject: Re: Concept Review: Recurrences (Part 1)  
Posted by cewrobel on Mon, 20 Feb 2017 20:57:23 GMT  
View Forum Message <> Reply to Message

Did we find out if sqrts are an acceptable form of the master theorem? Like is it acceptable for a to be sqrt(2) or b be sqrt(7)? Seems weird to have 1.41 sub problems lol.

---

Subject: Re: Concept Review: Recurrences (Part 1)  
Posted by lusth on Mon, 20 Feb 2017 23:36:14 GMT  
View Forum Message <> Reply to Message

Can't have 1.414... subproblems. Can have 1.414... as a subproblem size, though.