Consider this recurrence which is only defined for values of $n = 2^k$ for some integer $k \geq 1$:

$T(2) = 8$
$T(n) = 7 * n + T(n/2)$

1. Solve this recurrence using repeated substitution.

2. Try proving that your answer is correct by induction to see if you obtained a correct solution or not.

Lab 2 (for Monday Sept. 16) has been posted.
If you want extra problems for labs, let me know.

Assignment #1:
Written questions: due Fri. Sept. 20.
Programming questions: due Tues. Sept. 24 (connex).

If you did not fill out a schedule sheet last class, make sure you fill one out today.

Today I will go through slides we skipped from Lecture 2.

Office hours today:
12:30 or 1:30
Please let me know if you plan to come by.

The file itest.txt looks like this:

```
1 3
1 4

1 9
1 9

1 0
1 0

3 8 5 9
3 7 2 3

5 9 9 9 9 9
8 8 8 8 8 8 8 8

6 0 0 0 0 0 1
4 0 0 9 9

5 1 2 3 4 5
9 1 2 3 4 5 6 7 8 9
```

and so on…

Old versions of notepad are not displaying the new lines properly. There is a file itest2.txt now added to the bottom of programming assignment 1b that should work better for display. Or if you prefer, I have added the file contents to the bottom of the page so you can see it displayed properly.

Input/output should be from
STANDARD INPUT/STANDARD OUTPUT.
It is critical that you learn how to redirect standard I/O so that the I/O comes from a file.

Running from a command prompt:
To use the console for I/O:
java Test

To redirect the input to come from the file in.txt and send the output to out.txt:
java Test < in.text > out.txt

If you need help, stop by the consultants office.

Women in Engineering & Computer Science

COMPUTER SCIENCE REVEALED:

A clinic to help beginning programmers deepen their understanding of the art and science of programming.

- Are you a beginning programmer with little or no previous programming experience?
- Do you wish you understood a bit more about what goes on "under the covers" when you compile and run a Java program?
- Would you like to get a bit more practice debugging code?
- Would you like a few more "hands-on" exercises to help build your programming skills?

If you answered YES to any of these (or, if you'd just like to practice your programming a bit more, or transition over to Java from another language) come out to the free Computer Science Revealed Clinics. Everyone is welcome!

Our goal is to have a series of clinics throughout the term touching on different programming topics, all geared to the beginning programmer.

Our first clinic is on Friday September 13 @2:30, in ECS 348

Find us on Facebook

https://www.facebook.com/UvicWecs

The Clinic is hosted by the Women in Engineering and Computer Science group and is presented by CSC Grad Students, Faculty, and senior Undergrads – lots of hands-on exercises, lots of help, lots of fun.

Entrance is on a first come, first served basis. Looking forward to seeing you at the first Clinic!

Email: wecs@csc.uvic.ca
Website: wecs.csc.uvic.ca

Women in Engineering & Computer Science

3D PRINTING SEMINAR

Welcome back everyone!

Let's start the year off right by learning a little bit about 3D printing and watching the machine in action.

3D printing not enough?

Then come by for the free food!
The entrepreneurial company *College Pro* is providing PIZZA for the event.

When: Friday, September 13 @4:30
Where: ECS 660

Everyone welcome!

Find us on Facebook
https://www.facebook.com/UvicWecs

college pro

Email: wecs@csc.uvic.ca
Website: wecs.csc.uvic.ca

Consider this recurrence which is only defined for values of $n = 2^k$ for some integer $k \geq 0$:

$T(1) = 1$, $T(n) = n + T(n/2)$.

Last class I initially had the incorrect solution:

$T(2^k) = (k+1)\, 2^{k+1}$.

If we try to prove this is correct by induction:

[Base case]

$T(1) = T(2^0)$ is supposed to be 1 by definition of the recurrence.

If we plug k=0 into the formula we get:

$T(1) = T(2^0) = (0+1)\, 2^{0+1} = 2 \neq 1$.

The base case fails! If you have a correct solution to the recurrence, your formula should work for the base case.

Consider this recurrence which is only defined for values of $n = 2^k$ for some integer $k \geq 0$:

$T(1) = 1$, $T(n) = n + T(n/2)$.

The correct solution:

$T(2^k) = (k+1)\, 2^k$.

Exercise: Prove this is correct by induction.

We then expressed this in terms of n (the problem size).

Definition: if $n = 2^k$ then $\log_2(n) = k$.

Thus: $T(n) = T(2^k) = (k+1)\, 2^k = (\log_2(n) + 1)\, n$

$= n \log_2(n) + n \in O(n \log_2(n))$

Divide and Conquer

1. Divide the problem into two or more subproblems.
2. Solve the subproblems.
3. Marry the solutions.

This is one of the most common problem solving tactics and leads naturally to recursive algorithms.

Merge Sort- with linked lists

[Basis] If the list has size 0 or 1 it is already sorted so return.

[Divide] Otherwise, split the list into two lists, list1 and list2, of roughly equal sizes.

[Conquer] Sort list1 and list2 (recursively).

[Marry solutions] Merge list1 and list2 together to get the answer.

How can we split a list into two lists for our Mergesort?

One solution: evenOddSplit()

Side note: this is not the right way to do it on the assignment.

Learning objectives: Understanding of how to program linked lists in Java. To learn how to draw pictures of what a program is doing as it is executing (this will help you to more easily write correct code and to debug code which is not correct).

Review of linked lists