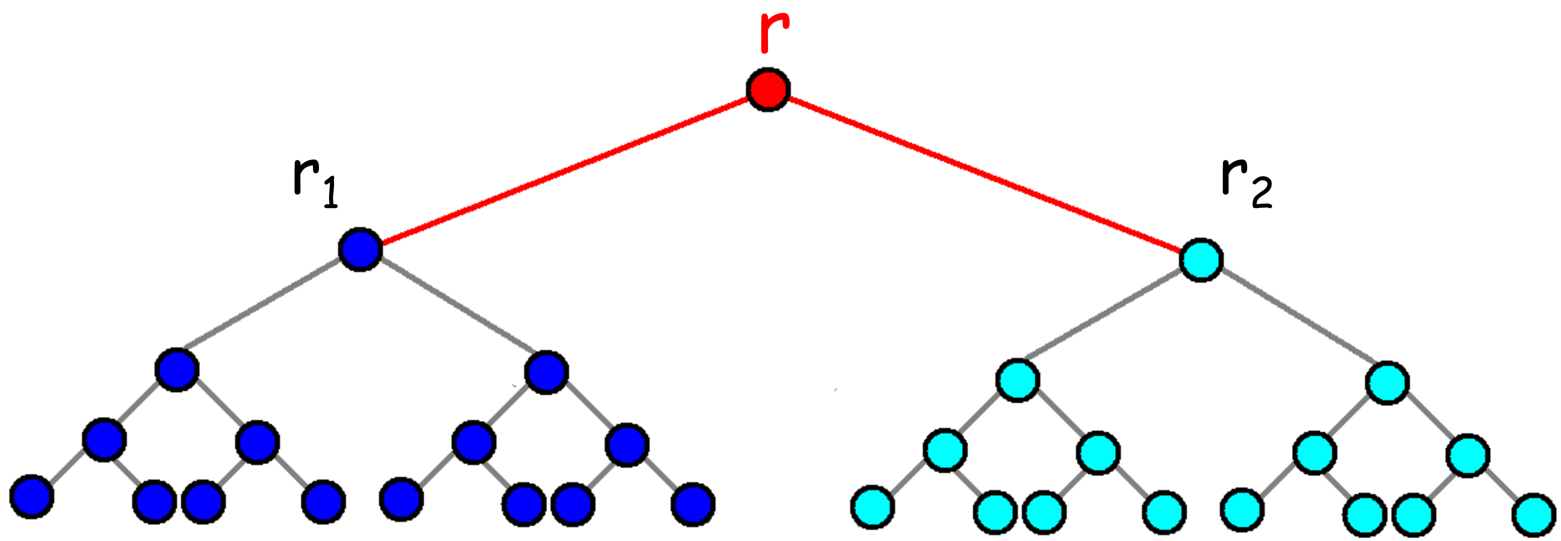
A complete binary tree of height 4:



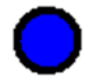

Problem of the day:

How many leaves does a complete binary tree of height h have? Propose a formula then prove it is correct by induction.

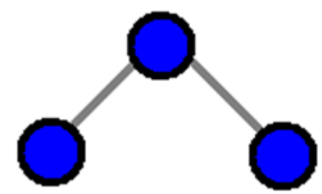
Use the induction definition of a complete binary tree presented in class.

Theorem: A complete binary tree of height h has 0 leaves when h = 0 and otherwise it has $2^h$ leaves.

Proof by induction.


Height 0

The complete binary tree of height 0 has one node and it is an isolated point and not a leaf. Therefore it has 0 leaves.


Height 1

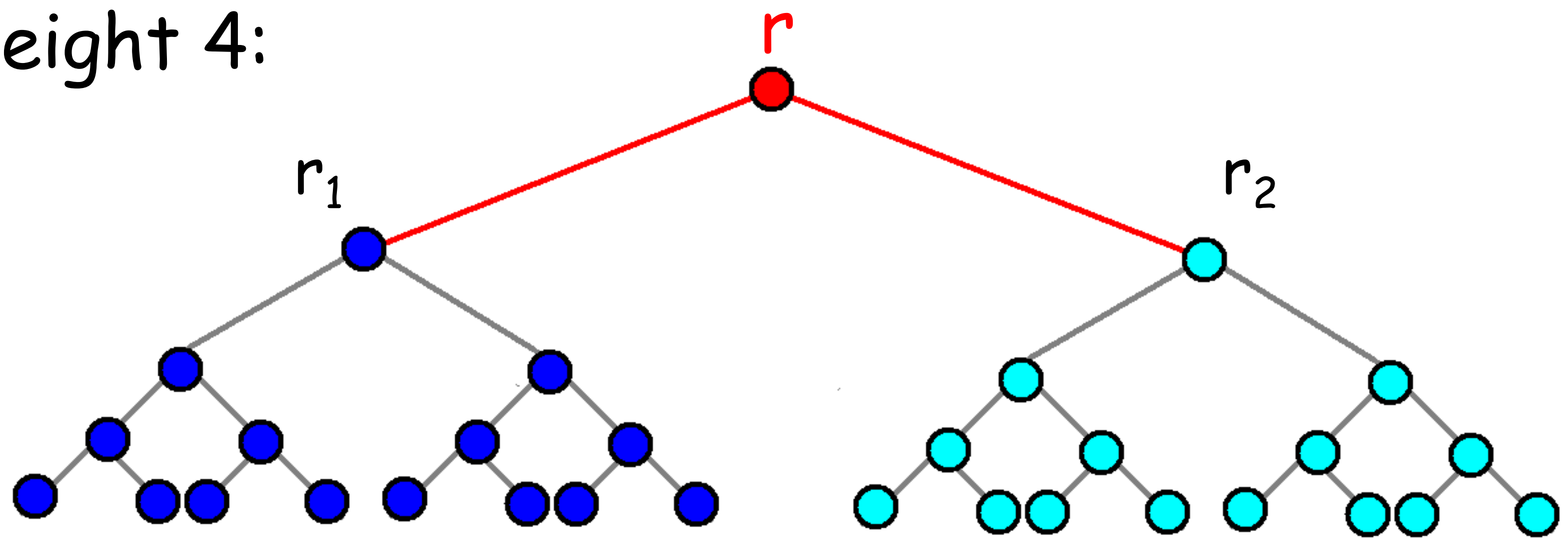
To make the induction get started,
I need one more case:
A complete binary tree of height 1 has two leaves.
The formula gives $2^1$ for height 1 and since $2^1 = 2$, the formula is correct for this case.

Assume that a complete binary tree of height h has $2^h$ leaves for $h \geq 1$.

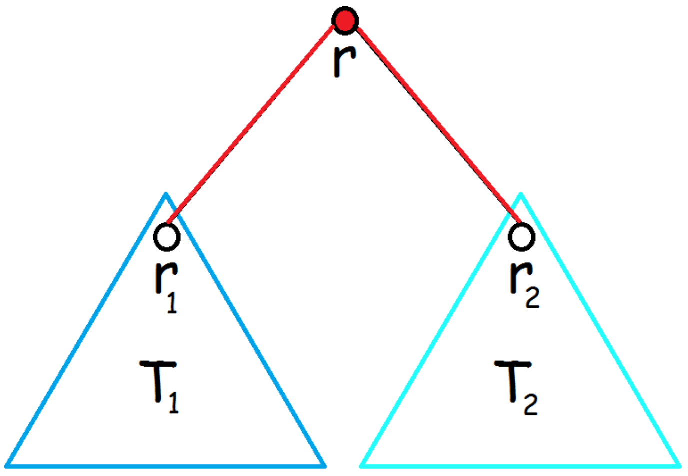
We want to prove that a complete binary tree of height h+1 has $2^{h+1}$ leaves.

A complete binary tree of height h+1 can be constructed by starting with two complete binary trees of height h, $T_1$ and $T_2$, that have root nodes $r_1$ and $r_2$ respectively, and adding a new root node r and edges $(r, r_1)$, and $(r, r_2)$.

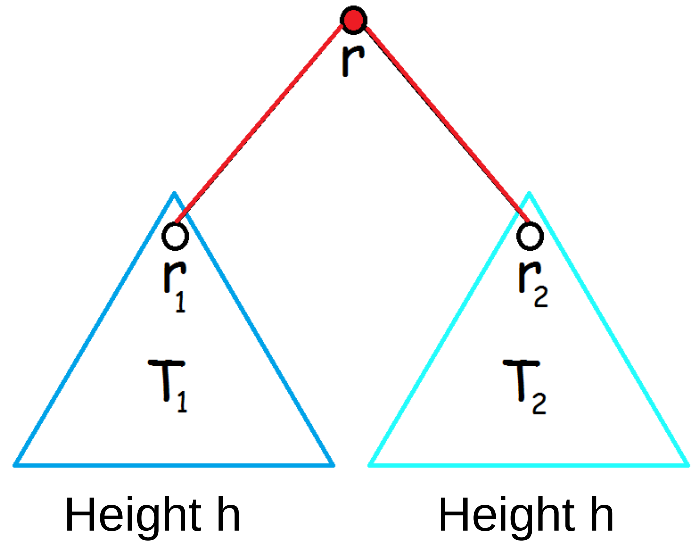
Example: A complete binary tree of height 4:




Generic picture of a complete binary tree of height h+1:



For $h \geq 1$, the leaves of this complete binary tree of height h+1 consist of the leaves of $T_1$ together with the leaves of $T_2$. Since $T_1$ and $T_2$ are complete binary trees of height h, by induction, each of these has $2^h$ leaves. So the total number of leaves is:



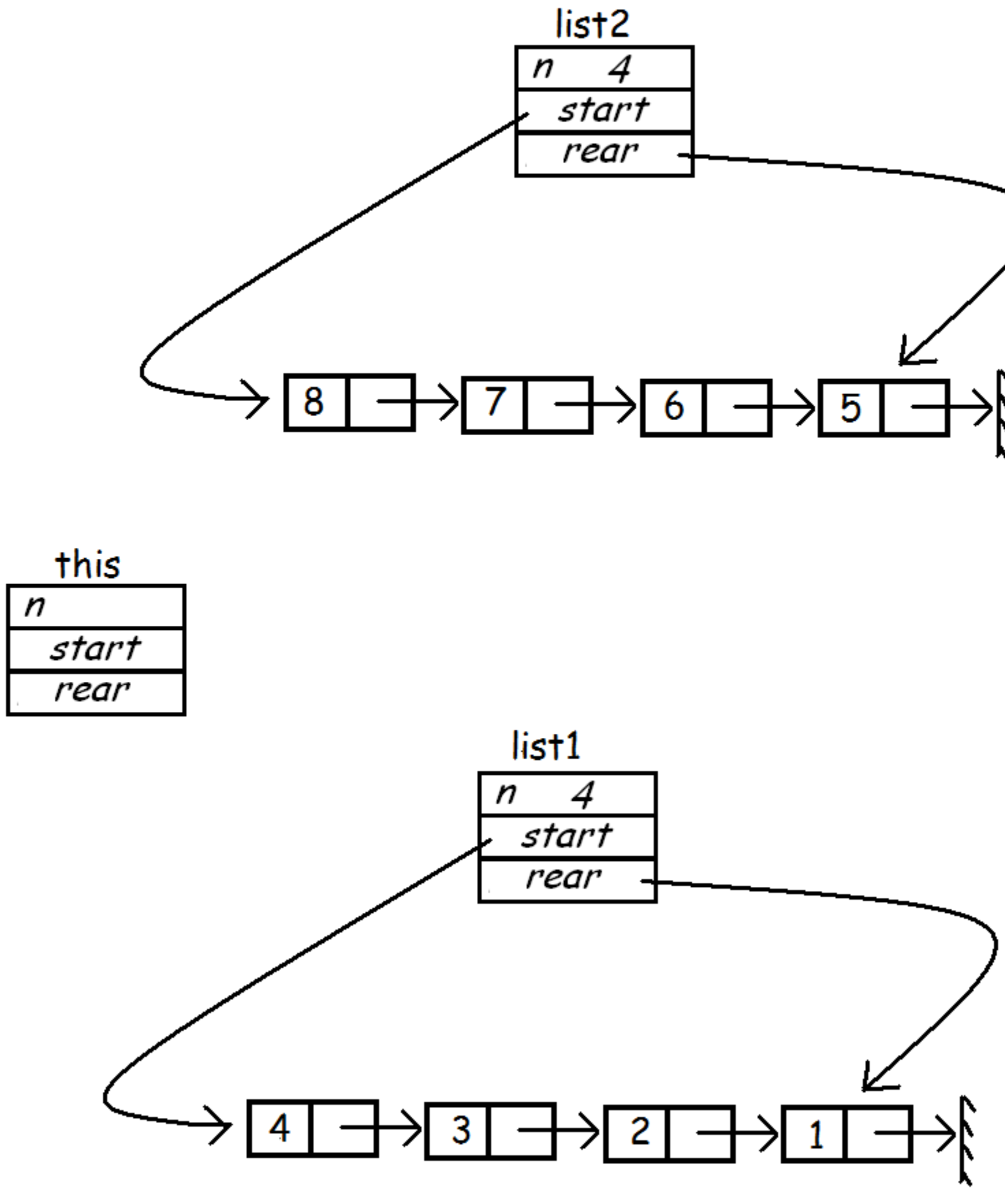

$2^h$ (number of leaves in $T_1$) + $2^h$ (number of leaves in $T_2$) $= 2^{h+1}$ as required.

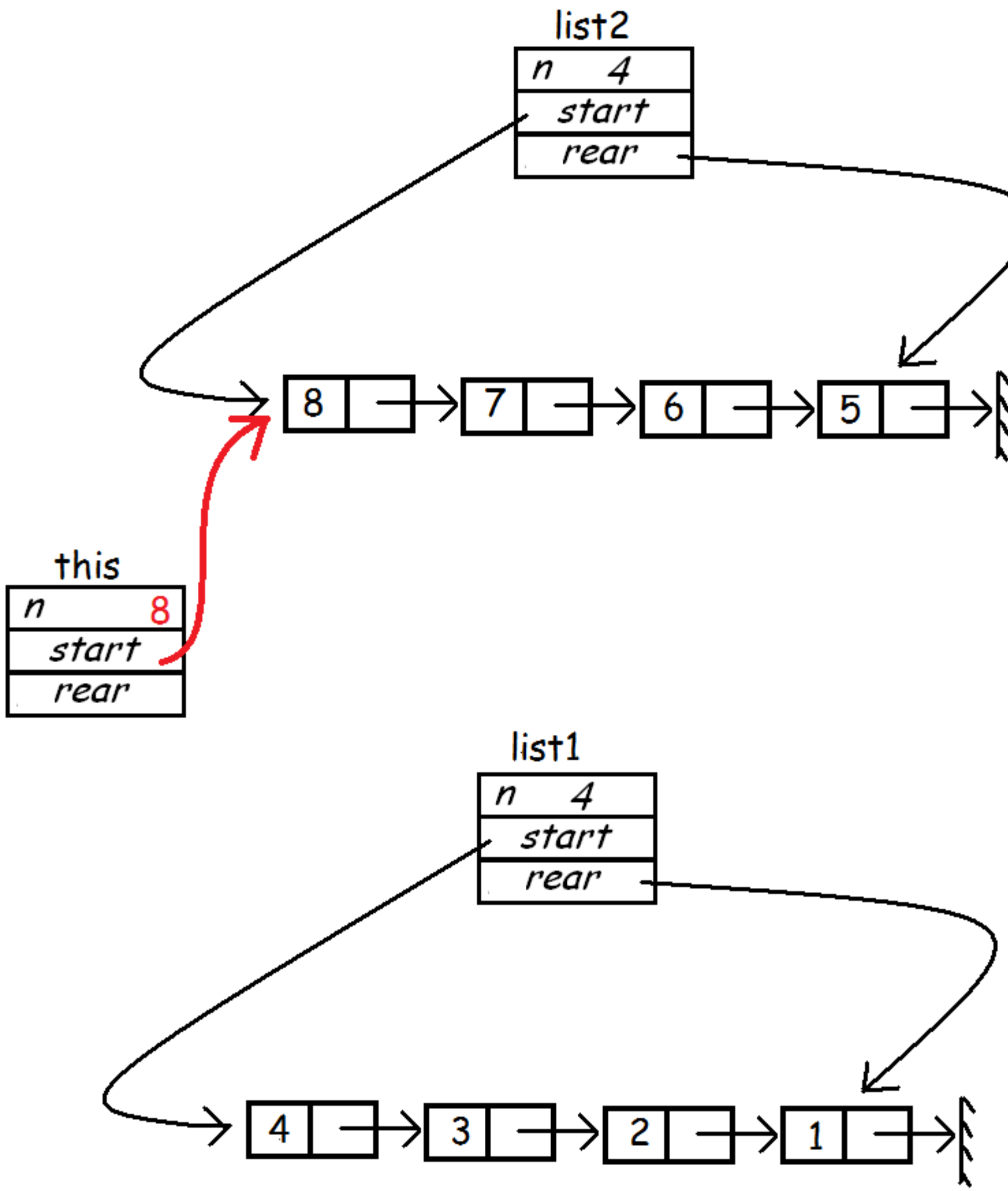
Recall: We want to prove that a complete binary tree of height h+1 has $2^{h+1}$ leaves.

Hint for writing code for linked lists:
As you are writing the code, try simultaneously executing it on a small example to make sure you are doing what you want to do.

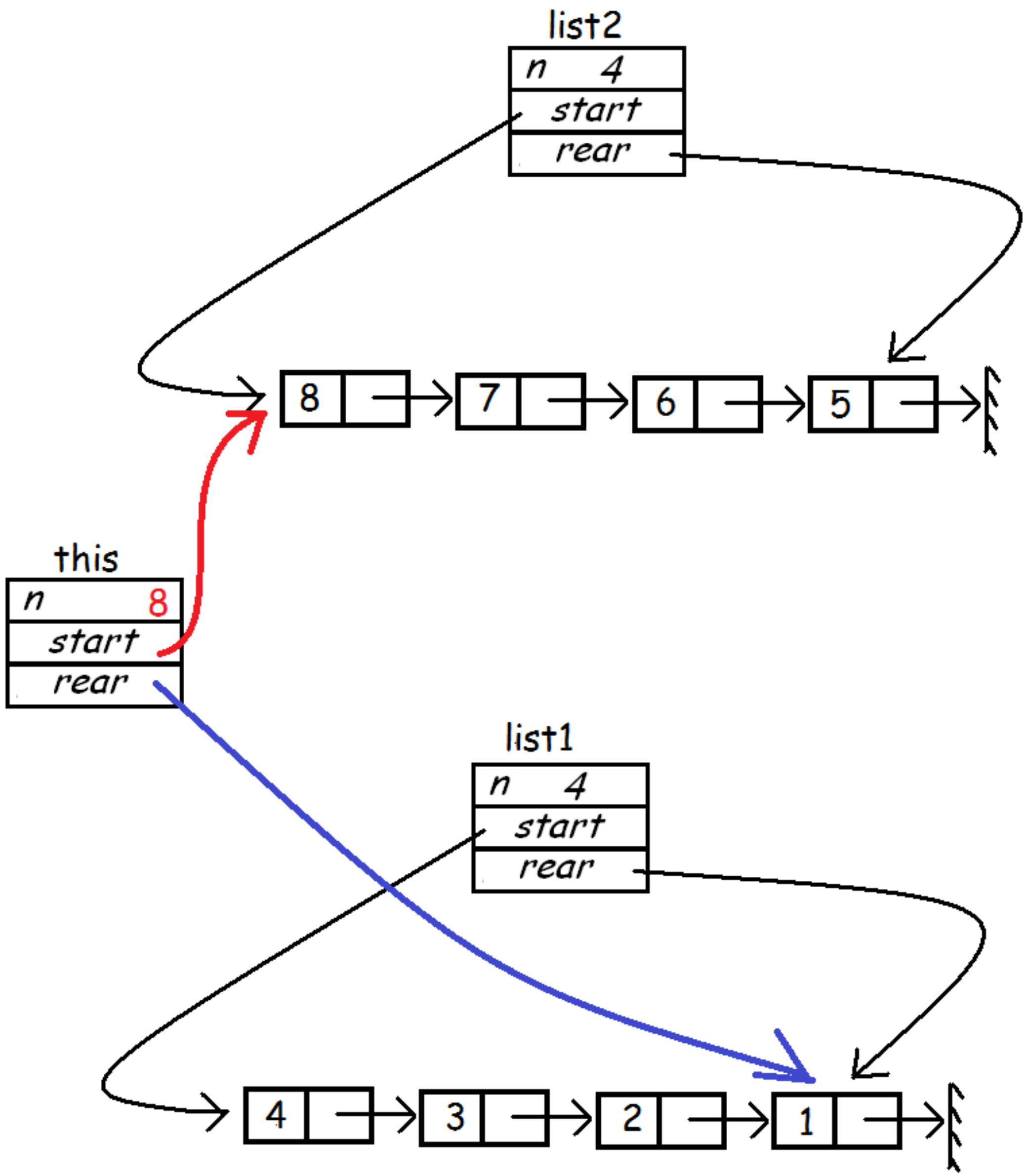
The example from last class originally had 1, 2, 3, 4, 5, 6, 7, 8 in the list.

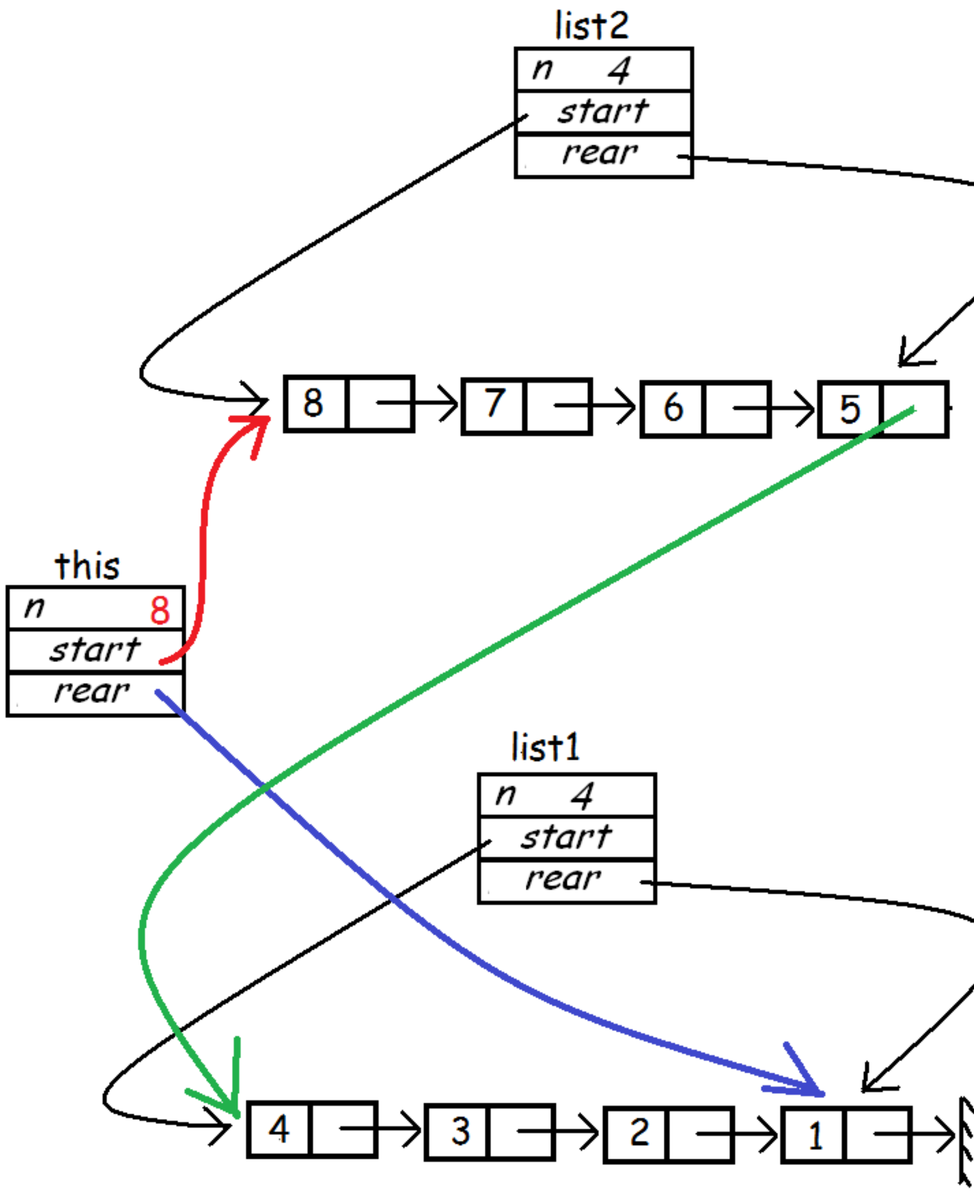
Consider the marriage step after the sublists with 1, 2, 3, 4 and 5, 6, 7, 8 are reversed.


list2
n 4
start
rear
8
7
6
5
this
n
start
rear
list1
n 4
start
rear
4
3
2
1



list2
n 4
start
rear
8
7
6
5
this
n 8
start
rear
list1
n 4
start
rear
4
3
2
1


list2
n 4
start
rear
8
7
6
5
this
n 8
start
rear
list1
n 4
start
rear
4
3
2
1


list2
n 4
start
rear
8
7
6
5
this
n 8
start
rear
list1
n 4
start
rear
4
3
2
1

After you have some code written, check for small examples and border cases that it works correctly on these.

Typical border cases:

Insertion at the beginning or end of a list.

List with size 0 or size 1.

For operations with two lists: if one or the other might be empty, make sure your code works correctly in these cases.

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

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

(a) What is T(16)? T(15)?

(b) Solve this recurrence using repeated substitution.

(c) Prove the answer is correct using induction.

To show your work when using repeated substitution, number your steps:

Step 0: The original recurrence for $T(n)$.

Step 1: The formula for $T(n)$ after one substitution into the RHS.

Step 2: The formula for $T(n)$ after two substitutions into the RHS.

...

DO not oversimplify by grouping terms together.

At step i, we want to be able to see what the ith term is and what the term is involving T.

Determine the general pattern for Step i:

Step i: The formula for T(n) after i substitutions into the RHS.

Determine at which step i the base case appears on the RHS of the formula, say at some step f.

Set i=f and then plug in the base case to get a formula for the recurrence relation that no longer has T in it.

Suppose the base case is changed to give a recurrence that is only defined for values of $n = 2^k$ for some integer $k \geq 3$:

$T(8) = 42$, and for $n \geq 16$, $T(n) = 1 + 2\,T(n/2)$.

What is the solution?