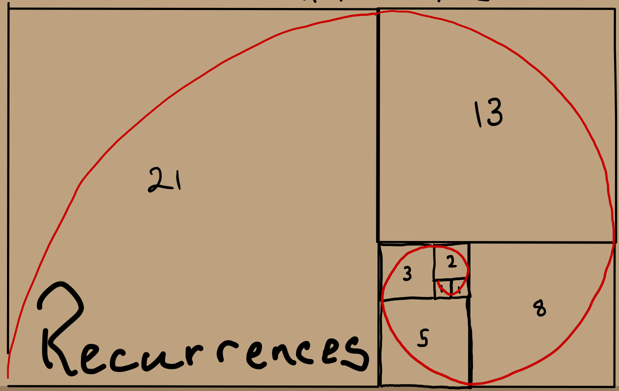
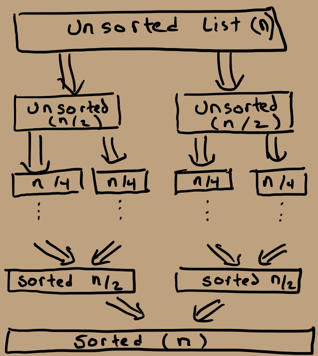
$$F_n = F_{n-1} + F_{n-2}$$



Recurrences



$$T(n) = 2T(n/2) + n$$

John Rachlin

CS 5002 : Discrete

Northeastern

Recurrence Relations (and how to solve them)

(A) Introduction

We have explored a few techniques for converting sequences into a closed form formula.

Sometimes it is easier to express a sequence as a recurrence relation: a formula for a_n expressed as a function of one or more prior terms.

A recursive definition is the recurrence relation plus initial conditions.

For example:

- Arithmetic sequences: $2, 4, 6, 8, \ldots$

  Closed form: $a_n = 2n$

  Recursive Defn: $a_n = a_{n-1} + 2 \quad ; a_1 = 2$

- Quadratic sequences: $1, 3, 6, 10, 15, \ldots$

  Closed form: $T_n = \dfrac{n(n+1)}{2}$ "Triangular numbers"

  Recursive Defn: $T_n = T_{n-1} + n \quad ; T_1 = 1$

- Geometric sequences: $3, 5, 9, 17, 33, 65, \ldots$

  Closed form: $a_n = 2^n + 1$

  Recursive Defn: $a_n = 2a_{n-1} - 1 \quad ; a_1 = 3$

* Using inductive hypothesis !

Can we verify this ?

$a_1 = 2^1 + 1 = 3$ ✓

$2a_{n-1} - 1 = 2(2^{n-1} + 1) - 1$

$= 2^n + 2 - 1$

$= 2^n + 1$

$= a_n$ ✓

Formal Proof By induction:

Prove : $\forall n: a_n = 2^n + 1$

Base case: $a_1 = 2^1 + 1 = 3$

Inductive Case :

Assume: $a_k = 2^k + 1$

Show : $a_{k+1} = 2^{k+1} + 1$

$= 2a_k - 1$ (by recurrence)

$= 2(2^k + 1) - 1 = 2^{k+1} + 1$ ✓

So verification may be easy, but we still want to be able to derive the closed form formula.

- Fibonacci Sequence: 0, 1, 1, 2, 3, 5, 8, 13, ...

  closed Form: ??? (we'll revisit this)

  Recurrence : $F_n = F_{n-1} + F_{n-2}$; $F_1 = 1$, $F_2 = 1$

Here the recursive definition is much easier to express. Note how small modifications to the definition give us new sequences:

Recurrence	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
$F_n = F_{n-1} + F_{n-2}$	0	1	1	2	3	5	8	13
	2	2	4	6	10	16	26	42
$F_n = 2F_{n-1} + F_{n-2}$	0	1	2	5	12	29	70	169
$F_n = F_{n-1} + 2F_{n-2}$	0	1	1	3	5	11	21	43

(B) Motivation

We are interested in deriving closed form formulas from the recursive definition.
Why? Some algorithms are recursive by nature, so we can use these techniques to analyze the asymptotic complexity of the algorithm.

(C) Solving via Iteration

$a_n = a_{n-1} + n \quad ; \quad a_0 = 4$

Sequence: $4, 5, 7, 10, 14, 19, \ldots$

Δ^1: $1 \quad 2 \quad 3 \quad 4 \quad 5, \ldots$

Δ^2: $1 \quad 1 \quad 1 \quad 1$ (constant)

So this is clearly a quadratic sequence.
From the definition: $a_n - a_{n-1} = n$

$$
\begin{array}{rrcl}
 & a_1 - a_0 & = & 1 \\
+ & a_2 - a_1 & = & 2 \\
+ & a_3 - a_2 & = & 3 \\
+ & a_4 - a_3 & = & 4 \\
\vdots & & & \vdots \\
+ & a_n - a_{n-1} & = & n \\
\hline
= & a_n - a_0 & = & \sum_{k=1}^{n} k = \frac{n(n+1)}{2}
\end{array}
$$

So, $a_n = \dfrac{n(n+1)}{2} + \underbrace{4}_{a_0}$

Let's solve it a different way, this time using iteration and substitution:

$a_n = a_{n-1} + n \quad ; \quad a_0 = 4$

$a_1 = a_0 + 1$

$a_2 = a_1 + 2$
$= (a_0 + 1) + 2$

$a_3 = a_2 + 3$
$= (a_1 + 2) + 3$
$= ((a_0 + 1) + 2) + 3$

$a_4 = a_3 + 4$
$= (((a_0 + 1) + 2) + 3) + 4$

$\vdots$

$a_n = ((\cdots(a_0 + 1) + 2) + 3) + \cdots) + n$
$= a_0 + \sum_{k=1}^{n} k = 4 + \dfrac{n(n+1)}{2}$

So the general strategy:

a) Repeatedly substitute

b) Look for an emerging pattern

c) Derive a formula from the pattern.

$$a_n = 3a_{n-1} + 2 \quad , \quad a_0 = 1$$

n :	0	1	2	3	4	...
a_n :	1	5	17	53	161	...

Differences: 4, 12, 36, 108

$$a_1 = 3a_0 + 2$$

$$a_2 = 3a_1 + 2 = 3(3a_0 + 2) + 2 = 3^2 a_0 + 2 \cdot 3 + 2$$

$$a_3 = 3a_2 + 2 = 3(3^2 a_0 + 2 \cdot 3 + 2) + 2 = 3^3 a_0 + 2 \cdot 3^2 + 2 \cdot 3 + 2$$

$$\vdots \qquad \vdots$$

$$a_n = 3a_{n-1} + 2 = \underbrace{3^n a_0}_{3^n} + \underbrace{2 \cdot 3^{n-1} + 2 \cdot 3^{n-2} + \cdots + 2 \cdot 3 + 2}_{2 \cdot (1 + 3 + 3^2 + \cdots + 3^{n-1})}$$

$$= 3^n + 2 \cdot \frac{(3^n - 1)}{2}$$

Since:

$$S = 1 + 3 + 3^2 + \ldots + 3^{n-1}$$

$$3S = 3 + 3^2 + \ldots + 3^{n-1} + 3^n$$

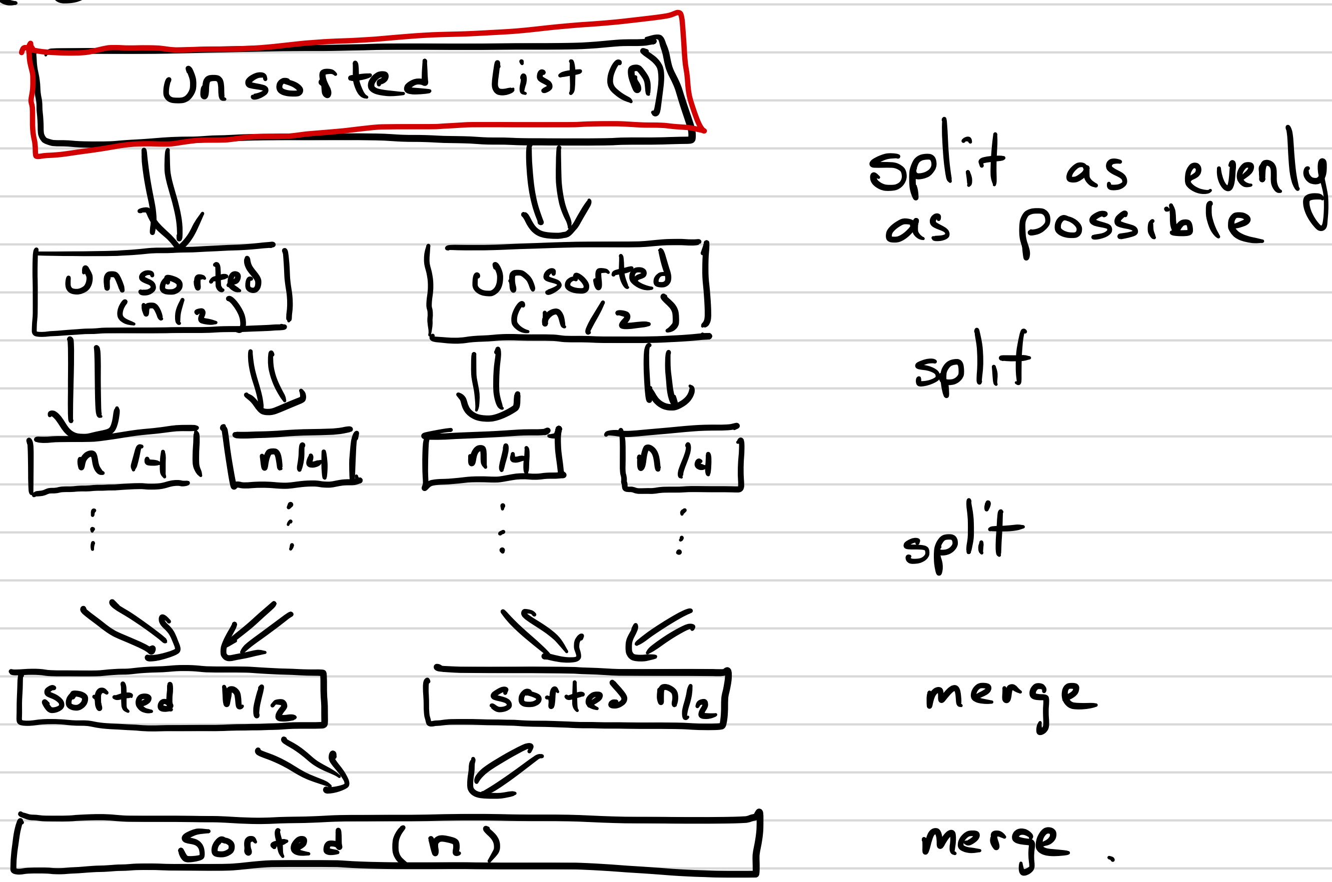
$$S - 3S = 1 - 3^n$$

$$-2S = 1 - 3^n$$

$$S = \frac{3^n - 1}{2}$$

$$\therefore \boxed{a_n = 3^n + 3^n - 1 = 2 \cdot 3^n - 1}$$

Merge Sort



The recursive definition is something like this:

$$T(n) = \text{Running time to merge sort } n \text{ numbers.}$$

$$= \underbrace{T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil)}_{\text{In case } n \text{ is odd}} + \underbrace{4n+2}_{\text{\# operations per merge}}$$

In case n is odd
e.g. $n = 17$:

$\lfloor 17/2 \rfloor = 8$

$\lceil 17/2 \rceil = 9$

\# operations per merge

But for asymptotic analysis, we can ignore floor/ceiling functions, constants, lower order terms.

$$T(n) = \begin{cases} 2\,T(n/2) + n & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Using iteration:

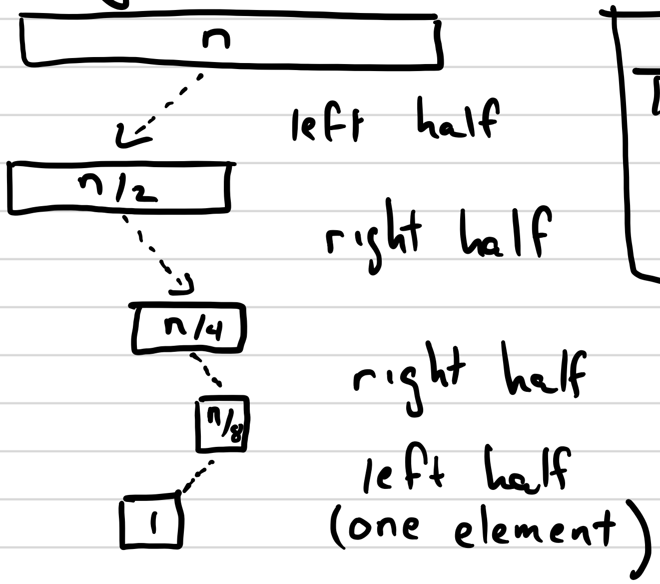
$$T(n) = n + 2T(n/2)$$
$$= n + 2\left(1/2 + 2T(n/4)\right)$$
$$= n + n + 4T(n/4)$$
$$T(n) = 2n + 4T(n/4)$$
$$= 2n\ 4(n/4 + 2T(n/8))$$
$$= 2n + n + 8T(n/8)$$
$$T(n) = 3n + 8T(n/8)$$
$$\vdots$$
$$T(n) = kn + 2^k T(n/2^k)$$

To eliminate the recursive term, we note that the algorithm bottoms out when $T(1) = 1$ $\therefore$

$$T(n/2^k) = 1 \quad \text{when} \quad n/2^k = 1$$
$$2^k = n$$
$$k = \log_2 n$$

$$\therefore T(n) = kn + 2^k T(n/2^k)$$
$$= \log_2 n \cdot n + 2^{\log_2 n}\, T(1)$$
$$= n \log_2 n + n \cdot 1 = O(n \log n)$$

Binary Search.



$$\boxed{\begin{aligned} T(n) &= 1 & n = 1 \\ &= T(n/2) + c & n > 1 \end{aligned}}$$

$$\begin{aligned} T(n) &= T(n/2) + c \\ &= \left(T(n/4) + c\right) + c \\ &= \left(\left(T(n/8) + c\right) + c\right) + c \\ &= T(n/2^k) + k \cdot c \end{aligned}$$

$T(1) = 1$ when $n/2^k = 1$ or $k = \log_2 n$

So we have:

$$\begin{aligned} T(n) &= T(1) + \log_2 n \cdot c = c \log_2 n + 1 \\ &= O(\log n) \end{aligned}$$

OPTIONAL

Characteristic Roots (Optional)

Given $a_n + \alpha a_{n-1} + \beta a_{n-2} = 0$

If r_1 and r_2 are distict roots of the characteristic polynomial

$$x^2 + \alpha x + \beta = 0$$

Then $\boxed{a_n = a r_1^n + b r_2^n}$

Or if only one root:

$$\boxed{a_n = a r^n + b n r^n}$$

where a and b are determined from initial conditions.

Example: $a_n = 7a_{n-1} - 10a_{n-2}$; $a_0 = 2, a_1 = 3$

n:	0	1	2	3	4	5	6
a_n:	2	3	1	-23	-171	-967	-5059

$$a_n - 7a_{n-1} + 10a_{n-2} = 0$$

Characteristic equation: $x^2 - 7x + 10 = 0$

Solve for x: $(x-2)(x-5) = 0$

So $x = 2$ or $x = 5$

$\therefore \boxed{a_n = a2^n + b5^n}$ $\quad a + b = 2 \Rightarrow 2a + 2b = 4$

$2a + 5b = 3$

$\Rightarrow 3b = -1$

$b = -1/3$

$a_n = \frac{7}{3}2^n - \frac{1}{3}5^n \Leftarrow a = \frac{7}{3}$; $b = -1/3$

Fibonacci Sequence

$$F_n = F_{n-1} + F_{n-2} \quad ; \quad F_0 = 0 \,,\ F_1 = 1$$

$$F_n - F_{n-1} - F_{n-2} = 0$$

$$x^2 - x - 1 = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad a = 1 \,,\ b = -1 \,,\ c = -1$$

$$= \frac{1 \pm \sqrt{1+4}}{2} = \frac{1 \pm \sqrt{5}}{2}$$

$$\varphi = \frac{1 + \sqrt{5}}{2} \quad ; \quad \psi = \frac{1 - \sqrt{5}}{2}$$

"phi" "psi"

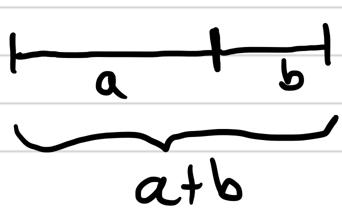
So $F_n = a\varphi^n + b\psi^n$ with $a + b = 0$

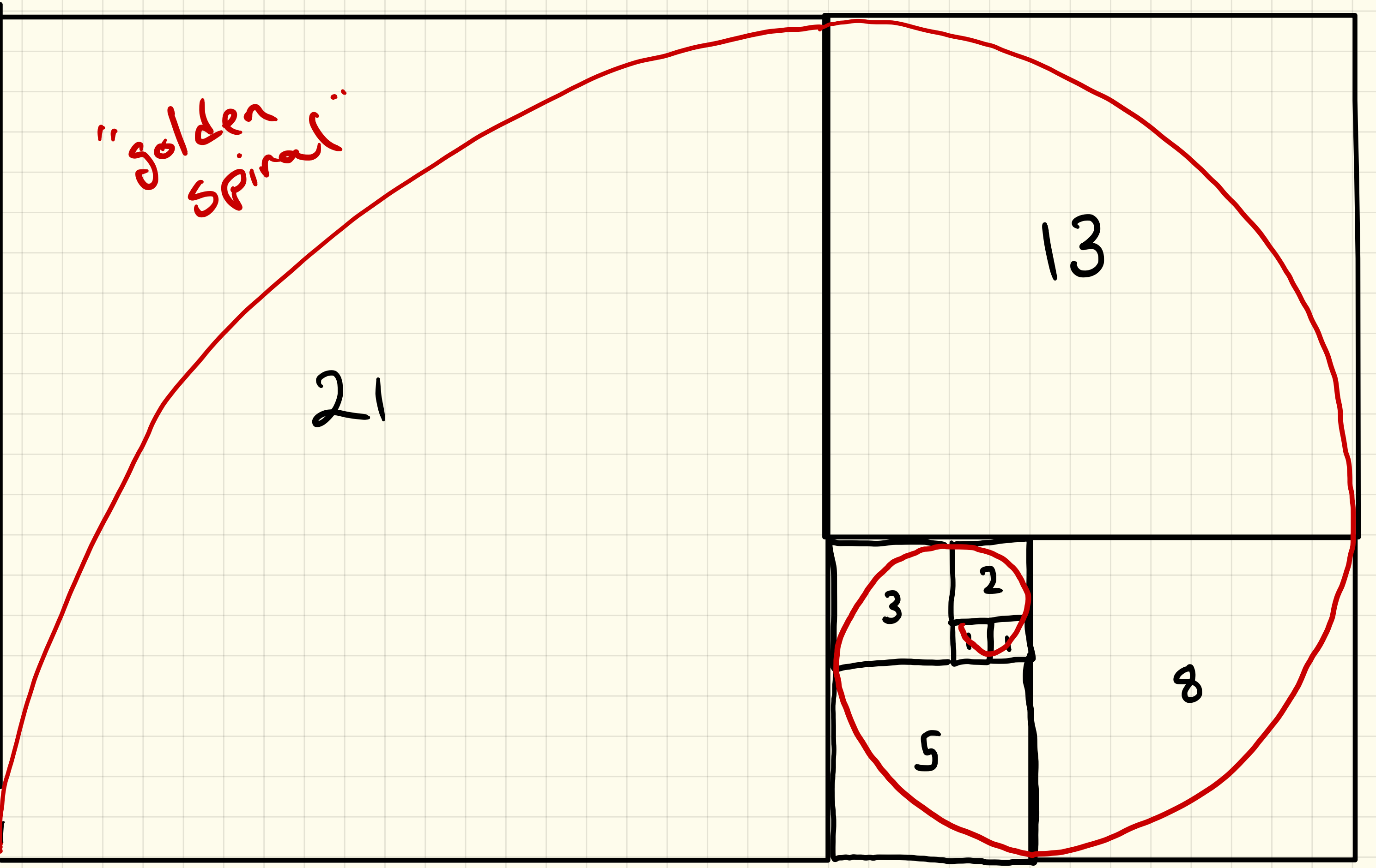
$a\varphi + b\psi = 1$

$$F_n = \frac{1}{\sqrt{5}}\varphi^n - \frac{1}{\sqrt{5}}\psi^n = \frac{\varphi^n - \psi^n}{\sqrt{5}}$$

φ = "The golden ratio" $\approx 1.6180339887\ldots$

$$\frac{a}{b} = \frac{a+b}{a}$$




"golden spiral"
21
13
8
5
3
2
1
1