assume the opposite of what you want

want equation to be true

CHAPTER 4: INDUCTION, RECURSION, AND RECURRENCES

4.1 Mathematical Induction

Final handshake

Smallest Counterexamples

Example: Prove the summation formula $0+1+2+...+n=\frac{n(n+1)}{2}$ for all integers $n \geq 0$ using a contradiction argument.

shouldn't have assume this

$A=B$

Proof: Suppose on the contrary that the formula is false, i.e. there exist positive integers n for which the formula fails to hold. Let $n=m$ be the smallest such case (called the *smallest counterexample*). This means that ON THE ONE HAND the formula is *false* for $n=m$:

$A \neq B$

"false case"

$$1+2+...+m \neq \frac{m(m+1)}{2}$$

$A \neq B$

principle of excluded middle

Observe that $m \neq 0$ since the formula is true for $n=0$:

$n \geq 0$ ← m can't be 0, has to be ≥ 1

$$0=\frac{0(0+1)}{2}$$

Thus, we have $m \geq 1$. Since m is the *smallest* counterexample, this also means that the formula is true for all $i<m$:

true for i

take a case smaller than m

$$1+2+...+i=\frac{i(i+1)}{2}$$

In particular, the formula is true for the case before m, namely $n=m-1$, which is a non-negative integer since $m \geq 1$. It follows that

$m-1 \geq 0$ → $m \geq 1$

$$1+2+...+(m-1)=\frac{(m-1)((m-1)+1)}{2}=\frac{(m-1)m}{2}$$

To obtain a contradiction, we add m to both sides of the formula for above and simplify to conclude that the formula ON THE OTHER HAND is *true* for $n=m$:

factor m

add m

$$1+2+...+(m-1)+m=\frac{(m-1)m}{2}+m \quad \left(=\frac{m^2-m}{2}+\frac{2m}{2}=\frac{m^2+m}{2}\right)$$

$A=B$

$$\therefore 1+2+...+(m-1)+m=\frac{m(m+1)}{2}$$

Since the formula can't be both true and false for the case $n=m$, this contradiction proves that there are NO counterexamples after all. Thus, the formula is true for all integers $n \geq 0$.

$p(n)=0+1+...+n=\frac{n(n+1)}{2}$

Proof by Smallest Counterexample: To prove the statement $\forall n \in \mathbb{N}(p(n))$, we proceed as follows:

natural #

opposite of what you want

if m is smallest case, anything smaller is true

1. Assume the existence of a smallest counterexample m so that ON THE ONE HAND $p(m)$ is *false*.

smaller case

2. Using the fact that $p(m')$ must be true for all $m'<m$, choose a specific value for m', say $\boxed{m'=m-k}$ (e.g. $m-1$) where the choice of k depends on the problem.
3. Prove the implication $p(m-k) \Rightarrow p(m)$ so that ON THE OTHER HAND $p(m)$ is *true*.
4. Conclude by contradiction that $p(n)$ is true for all $n \in \mathbb{N}$.

contradictory assumption

Example: Use the technique of proof by smallest counterexample to demonstrate Euclid's Division Theorem (EDT), i.e. given any positive integer *n*, any non-negative integer *m*, there exists unique non-negative integers *q* and *r* such that $m = nq + r$ with $0 \le r < n$.

Proof: Assume on the contrary that EDT fails to hold, i.e. for some given integer *n*, there are values *m* for which no such *q* and *r* exist. We then set *m* to be the smallest counterexample. Observe that $m \ne 0$ since EDT holds for 0:

$$0 = n \cdot 0 + 0 \qquad (q = 0, r = 0)$$

With *m* being the smallest counterexample, this means given any value $m' < m$, EDT is true for m', i.e. there exist q' and r' satisfying $m' = nq' + r'$ with $0 \le r' < n$. To obtain a contradiction, we set $m' = m - n$. Then there exist q' and r' satisfying $m - n = nq' + r'$ with $0 \le r' < n$. We solve for *m* to obtain

$$m = nq' + r' + n = n(q' + 1) + r'$$

Now set $q = q' + 1$ and $r = r'$. This shows that such *q* and *r* do exist for *m*, which contradicts the assumption that no such *q* and *r* exist. It follows that there are NO such counterexamples for *m*. Hence, EDT holds for all non-negative integers *m*.

Mathematical Induction versus Scientific Induction (Traps and Pitfalls):

Example: Consider the statement: The product of the first n primes plus one is always prime.

Scientific Induction: We conduct an experiment by calculating the following values:

2+1=3 (prime)
2*3+1=7 (prime)
2*3*5+1=31 (prime)
2*3*5*7+1=211 (prime)

checks each case one at a time

Since the current evidence supports the statement, we hypothesize that it is true. We then continue to perform more experiments to see if the statement continues to remain true:

Divisors of 3 = {3, 1}
Divisors of 7 = {7, 1}
Divisors of 31 = {31, 1}}
Divisors of 211 = {211, 1}
Divisors of 2*3*5*7*11+1=2311 = {2311, 1}
Divisors of 2*3*5*7*11*13+1=30031 = {59, 509, 1}

think about what values of n contains $(n \in \mathbb{N})$

But alas we found a counterexample (30031) and so the statement is false.

Example: Consider the statement: The sum of the first *n* odd integers equals n^2.

needs explanation

Again, scientific induction calls us to perform experiments:

$n=1: \ 1 = 1^2$ ✓

$n=2: \ 1+3 = 2^2$, $4 = 4$ ✓

$n=3: \ 1+3+5 = 3^2$, $9 = 9$ ✓

knocking down first domino

$1 = 1^2$

$1+3 = 2^2$

$1+3+5 = 3^2$

$1+3+5+7 = 4^2$

...

$1+3+5+...+(2n-1) = n^2$] what you want to prove

$n=3$: $2n-1 = 2\cdot 3 - 1 = 5$ ✓

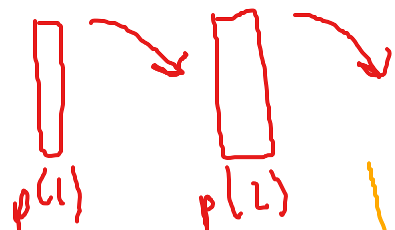
* extra term, why you add to both sides

last term here in common w/ $p(n-1)$

prove it true

Since we know in hindsight that the statement is true, then we would have experiment indefinitely. In general, this cannot be considered an effective direct method of proof since there are infinitely many cases to consider.

Domino Induction (Analogy) True ⇒ Domino falls



Principle of Mathematical Induction: If the two statements

I. (Base case) $p(b)$ smallest case in prob

II. (Inductive step) $p(n-1) \Rightarrow p(n)$ for all $n > b$

are both true, then $p(n)$ is true for all integers $n \ge b$.

previous case ⇒ next case

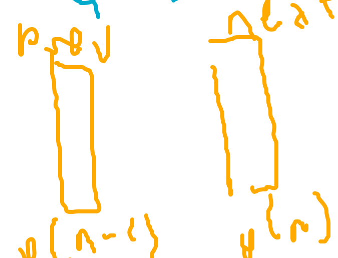
Example: Prove that the sum of the first n odd natural numbers equals n^2, i.e.

n = 1, 2, 3

$$p(n): \quad 1+3+5+...+(2n-1) = n^2$$

for $n \ge 1$ by mathematical induction.

$p(4)$: $1+3+5+7 = 4^2$ ✓



Proof: after checking b = 0 separately

I. The base case $b=1$ is true since

$$1 = 1^2$$ ✓

II. Suppose $p(n-1)$ is true:

$$1+3+5+...+(2(n-1)-1) = (n-1)^2$$

We now demonstrate that $p(n)$ must also be true. Add $2n-1$ to both sides of the equation above:

$$1+3+5+...+(2(n-1)-1)+(2n-1) = (n-1)^2+(2n-1) = (n^2-2n+1)+(2n-1)$$ algebra

or equivalently, simplify

$$\therefore 1+3+5+...+(2n-1) = n^2$$

This proves that $p(n)$ is true. Thus, $p(n)$ is true for all positive integers n by the Principle of Mathematical Induction. make domino effect happen

NOTE: Observe that the argument is essentially the same as proof by contradiction via smallest counterexample. Indeed, the proof for Mathematical Induction relies on this exact argument.

(I) (II)
$p(1) \Rightarrow p(2)$
inference

if prev true, next must be true too

Example: Prove that $2^n > n^2$ for all integers $n \ge 5$.

Proof:

I. The base case $n=5$ is clearly true:

$$2^5 = 32 > 25 = 5^2$$

II. Suppose case $n-1$ is true:

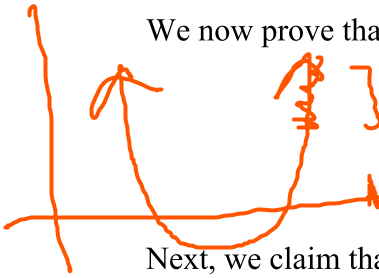
$x^{a-b} = \frac{x^a}{x^b}$

Corresponding

$$2^{n-1} > (n-1)^2 \longrightarrow p(n) \quad 2^n > n^2$$

need to prove

We now prove that case n must also also true by manipulating the equation above as follows:



have @ 5

$5 \le n$

$$\frac{2^n}{2} > (n-1)^2 = n^2 - 2n + 1$$
$$\Rightarrow 2^n > 2n^2 - 4n + 2$$
$$\Rightarrow 2^n - n^2 > n^2 - 4n + 2 \qquad A > B$$

Next, we claim that $n^2 - 4n + 2 \ge 0$ for all $n \ge 5$. This is true because (work backwards to obtain insight)

$f(n) = n^2 - 4n + 2$
- find where evaluates to 0

$$n \ge 5$$
$$\Rightarrow n \ge 2 + \sqrt{2} \, (\approx 3.414)$$
$$\Rightarrow n - 2 \ge \sqrt{2}$$
$$\Rightarrow (n-2)^2 \ge 2$$
$$\Rightarrow n^2 - 4n + 4 \ge 2$$
$$\therefore n^2 - 4n + 2 \ge 0$$

want $2^n - n^2 > 0$

need $B > C$

$A > C$

By the transitive property of inequalities, we conclude that $2^n - n^2 > 0$, or equivalently, $2^n > n^2$. This proves that case n is true. Hence, the inequality must be true for all integers $n \ge 5$.

regular: $p(n-1) \Rightarrow p(n)$

NOTE: In general, we have $n - 2 \ge \sqrt{2} \Rightarrow (n-2)^2 \ge 2$, but $(n-2)^2 \ge 2 \not\Rightarrow n - 2 \ge \sqrt{2}$.

Principle of Strong Induction: If the two statements
I. Base case: $p(b)$
II. Inductive step: $\underbrace{p(b) \wedge p(b+1) \wedge \ldots \wedge p(n-1)}_{\text{inductive assumption}} \Rightarrow \boxed{p(n)}$ for all $n > b$
are both true, then $p(n)$ is true for all integers $n \ge b$.

every case before that

prove

some can be proven by regular

Example: Prove the following statement using strong induction: Every integer $n \ge 2$ is divisible by a prime number.

cuz exactly

Proof:
I. Base case: Clearly $b = 2$ is divisible by a prime number, namely itself (2 is prime).
II. Inductive step: Suppose every integer between 2 and $n-1$ is divisible by a prime. We now prove that n is divisible by a prime number. Now consider two cases:
CASE I: n is prime. In that case n is divisible by itself (being prime) and we are done.
CASE II: n is composite, i.e. $n = ab$, where $2 \le a, b \le n-1$. By the (strong) inductive step, we have that a, being a number between 2 and $n-1$, is divisible by some prime p. It follows that $\boxed{n = ab \text{ is also divisible by } p}$. Hence, by mathematical induction all integers $n \ge 2$ are divisible by a prime number.

covers all cases b/c a # is prime or composite

a is divisible by p → $a = p \cdot q$ → $a \cdot b = pqb$

30 is divisible by 5 → $30 = 5 \cdot 6$

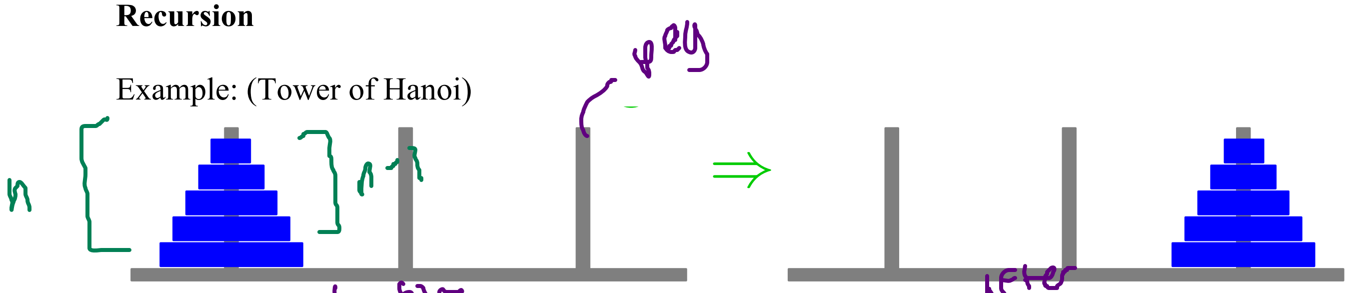
$n = p(qb)$
$n = pr$
n is divisible by p

sequence = string of values (many #s)
→ $a_1, a_2, a_3 \ldots$
$m(0), m(1), m(2)$

4.2 Recursion, Recurrences and Induction

Recursion

Example: (Tower of Hanoi)



Problem: Determine an algorithm to move n disks on Peg 1 to Peg 3. Disks can be moved to any of the pegs, but only one at a time, and can be stacked on top of each other, but a larger disk can never placed on top of a smaller disk. How many moves are needed?

Solution: Here is a recursive algorithm for solving the problem:
  Step 1. Move the top $n-1$ disks on peg 1 to peg 2.
  Step 2. Move disc n to peg 3.
  Step 3. Move the $n-1$ discs on peg 2 to peg 3.

n = # of disks
M(n) = # of moves

Let $M(n)$ denote the number of moves needed to move n disks from peg 1 to peg 3. Then it is clear that $M(1)=1$. Next, each step in the algorithm above requires the following number of moves:
  Step 1. $M(n-1)$ moves
  Step 2. 1 move
  Step 3. $M(n-1)$

add

strong induction used in Fibonacci

Thus, the total number of moves needed is given recursively by

(recurrence) R $\boxed{M(n)=2M(n-1)+1}$

for $n>1$. Note that we can iterate this recurrence to obtain the next few values of $M(n)$:

n	M(n)
1	1
2	3
3	7
4	15

not always better

smallest value for n = 1

$$M(2)=2M(1)+1=2(1)+1=3$$
$$M(3)=2M(2)+1=2(3)+1=7$$
$$M(4)=2M(3)+1=2(7)+1=15$$

P need to prove

We infer from the pattern that the general formula for $M(n)$ is given by $M(n)=2^n-1$, which we prove using induction:

explicit/direct formula

must verify formulas

I. Base case: the formula clear holds for $n=1$: $M(1)=2^1-1=1$

II. Inductive step: suppose the formula holds for $n-1$:

I $p(n-1)$ → $\boxed{M(n-1)=2^{n-1}-1}$ (Assume true) inductive assumption

To prove that the formula holds for n, we substitute the formula above into the recurrence, which holds for all n:

A R I B

R + I = P

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

regular

Hence, the formula must be true for all positive integers n by mathematical induction.

for M(n) problem

First-Order Linear Recurrences

Definition: A *first-order linear recurrence* for a sequence $T(n)$ is one of the form

1st order uses 1 prev term
M(n) = 2M(n-1)+1

2nd order
F(n) = F(n-1)+F(n-2)

linear = multiply and add

$y \quad f(n) \quad g(n)^x$

$y = 2x+1$

$$\boxed{T(n) = f(n)T(n-1) + g(n)}$$

Recurrence

where $f(n)$ and $g(n)$ are two given sequences.

T(n) = explicit formula

Examples:

(a) $T(n) = 2T(n-1)+1$

(b) $T(n) = T(n-1)+n$ (Explicit formula for $T(n)$?) add variable $f(n)=1$, $g(n)=n$

(c) $T(n) = nT(n-1)$ (Explicit formula for $T(n)$?) multiply variable $f(n)=n$, $g(n)=0$

(d) Let $S(n)$ denote the number of possible subsets that can be constructed from a set A that contains n elements, e.g. $A = \{1,2,...,n\}$. Find a recurrence for $S(n)$ in terms of $S(k)$ where $k < n$.

Solution: If $n = 0$, then $A = \varnothing$ (empty set). In that case, $S(0) = 1$ since there is only one subset, namely A itself. To determine a recurrence, we examine the subsets in the case $n = 2$ and $n = 3$:

Subsets of $A = \{1,2\}$	Subsets of $A = \{1,2,3\}$
$\varnothing$, $\{1\}$, $\{2\}$, $\{1, 2\}$	Type I: $\varnothing$, $\{1\}$, $\{2\}$, $\{1, 2\}$ Type II: $\{\mathbf{3}\}$, $\{1, \mathbf{3}\}$, $\{2, \mathbf{3}\}$, $\{1, 2, \mathbf{3}\}$

We infer from the table above that the subsets of $A = \{1,2,...,n\}$ can be divided into two types:
I. Subsets that do not contain the largest element n
II. Subsets that do contain the largest element n
But observe that the subsets of type I are exactly the subsets of $\{1,2,...,n-1\}$. Moreover, the number of subsets of each type are equal each other, namely $S(n-1)$, because of the one-to-one correspondence

$$\overset{\text{Type I}}{\text{subset } B} \leftrightarrow \overset{\text{Type II}}{\text{subset } B \cup \{3\}}$$

Thus,

$$S(n) = 2S(n-1)$$

Exercise: Find and prove an explicit formula for $S(n)$.

NOTE: The set of all subsets of A is called the *power set* of A and denoted by 2^A.

Iterating a Recurrence

Next = r · previous + a

Given a recurrence of the form $T(n) = rT(n-1) + a$, where r and a are constants, we can iterate backwards to obtain an explicit formula:

$T(n-1) = r \cdot T(n-2) + a$

$$T(n) = rT(n-1) + a$$

sub + distrib

$$= r[rT(n-2)+a] + a = r^2T(n-2) + ra + a$$

next and sub / distrib again

$$= r^2[rT(n-3)+a] + ra + a = r^3T(n-3) + r^2a + ra + a$$

$$= r^3[rT(n-4)+a] + r^2a + ra + a = r^4T(n-4) + \underbrace{r^3a + r^2a + ra + a}$$

From the pattern above we infer the formula coefficient powers of r things

$$T(n) = \boxed{r^nT(0) + r^{n-1}a + r^{n-2}a + ... + ra + a} = r^nT(0) + a\boxed{\sum_{i=0}^{n-1} r^i}$$

NOTE: We can also iterate forwards to infer the same formula:

$T(n) = r^5 \cdot T(n-5) + r^4 a$ Not more efficient

:

- factor out a
- r to some power
- range for i ?

$= r^n(T(n-n)) + r^{n-1}a + r^{n-2} + \dots r^1 a + r^0 a$

$T(1) = rT(0) + a$

$T(2) = rT(1) + a = r[rT(0) + a] + a = r^2T(0) + ra + a$

$T(3) = rT(2) + a = r[r^2T(0) + ra + a] + a = r^3T(0) + r^2a + ra + a$

Geometric Series (Sum)

similar to deriving handshake formula

To find an efficient formula for the geometric sum

$$S = 1 + r + r^2 + \dots + r^{n-1}$$

we first multiply through by r:

$$rS = r + r^2 + \dots + r^n$$

We then subtract the two equations to obtain a sum that telescopes:

$O(r^n), r > 1$

$O(1), r < 1$

further b out

$$S - rS = (1 + r + r^2 + \dots + r^{n-1}) - (r + r^2 + \dots + r^n)$$

$$\Rightarrow (1-r)S = 1 - r^n$$

$$\therefore S = \frac{1-r^n}{1-r}$$

survivors of cancellation

Lemma: (Geometric series) If $r \neq 1$, then

$$\boxed{G(n) \equiv \sum_{i=0}^{n-1} r^i = \frac{1-r^n}{1-r}}$$

$= \begin{cases} O(r^{n-1}), & r > 1 \\ O(1), & r < 1 \end{cases}$

Theorem: If $T(n) = rT(n-1) + a$, $T(0) = b$, and $\boxed{r \neq 1}$, then for all non-negative integers n,

$$O(?) = \boxed{T(n) = r^n b + a\frac{1-r^n}{1-r}}$$

magical formula

$= r^n b + a(G(n))$

$= r^n b + a\left(\underbrace{\frac{1}{1-r}}_{A} - \underbrace{\frac{r^n}{1-r}}_{B}\right)$

NOTE: This theorem can be proven using induction.

texmoush

Example: Solve the recurrence $T(n) = 4T(n-1) + 5$, $T(0) = 2$, by finding an explicit formula for $T(n)$ and use it to compute $T(10)$. NOTE: What if the initial value given was $T(1) = 2$.

can't use theorem

AB

Solution: Applying the theorem above to our recurrence with $r = 4$, $a = 5$, and $b = 2$ yields

which is bigger, A or B?

depends on r

$$T(n) = 4^n \cdot 2 + 5\frac{1-4^n}{1-4} = 2 \cdot 4^n - \frac{5}{3}(1-4^n)$$

$$= \frac{11}{3} \cdot 4^n - \frac{5}{3}$$

efficient answer (direct formula)

We now easily compute $T(10)$:

$$T(10) = \frac{11}{3} \cdot 4^{10} - \frac{5}{3} = 3844777$$

If $r = 1$,
$T(n) = T(n-1) + a$
Iterate

Lemma: Suppose $r > 0$ and $r \neq 1$. Denote by $t(n)$ the largest term in the geometric series $G(n)$. Then $G(n) = \Theta(t(n))$.

Proof: We first prove $G(n) = O(t(n))$ by considering two cases:

$T(0) = b$
$T(1) = T(0) + a$
$= b + a$

n (a) is dominant

$O(n) \leftarrow b + n \cdot a$

$f(n) \le c \cdot g(n) \longrightarrow f = O(g)$

I. $r > 1$. Then the largest term in $G(n)$ is r^{n-1}. It follows from the previous lemma that

b term is larger

$$G(n) = \frac{1-r^n}{1-r} = \frac{r^n - 1}{r-1} \quad = \underbrace{\frac{r}{r-1}}_{B} - \underbrace{\frac{1}{r-1}}_{A}$$

$$\le \frac{r^n}{r-1} = r^{n-1} \cdot \frac{r}{r-1}$$

$$\le c \cdot r^{n-1} \quad \left(c = \frac{r}{r-1} \right)$$

$$\therefore G(n) = O(r^{n-1})$$

minus out negligible

B dominant, A negligible

II. $r < 1$. Then the largest term in $G(n)$ is 1. It follows again from the previous lemma that

$$G(n) = \frac{1-r^n}{1-r} = \frac{1}{1-r}$$

$$\le c \cdot 1 \quad \left(c = \frac{1}{1-r} \right)$$

$$\therefore G(n) = O(1)$$

A is dominant
B is negligible

G(n) doesn't grow any bigger than an n function

We leave the proof of $t(n) = O(G(n))$ as an exercise. This proves $G(n) = \Theta(t(n))$.

Formulas for First-Order Linear Recurrences

$T(n) = r \cdot T(n-1) + a$
$T(0) = b$

Theorem: Let $T(n)$ be defined by the recurrence

$$T(n) = \begin{cases} rT(n-1) + g(n) & \text{if } n > 0, \\ a & \text{if } n = 0. \end{cases}$$

not always able to get explicit formula

Then

$$T(n) = ar^n + \underbrace{\sum_{i=0}^{n-1} r^i g(n-i)}_{\text{backwards}} = ar^n + \underbrace{\sum_{i=1}^{n} r^{n-i} g(i)}_{\text{forwards}}$$

plug in to back or forwards

Proof: (Induction)

Example:
(a) Solve the recurrence $T(n) = 4T(n-1) + 2^n$ with $T(0) = 6$. $\quad r = 4,\ g(n) = 2^n,\ a = 6$
(b) Solve the recurrence $T(n) = 3T(n-1) + n$ with $T(0) = 10$. $\quad r = 3,\ g(n) = n,\ a = 10$

Solution:
(a) Applying the theorem above yields

simplify w/ algebra

$g(n-i) = 2^{n-i}$

$2^{n-i} = \frac{2^n}{2^i}$

$$T(n) = \boxed{6 \cdot 4^n + \sum_{i=0}^{n-1} 4^i \cdot 2^{n-i}} = 6 \cdot 4^n + 2^n \sum_{i=0}^{n-1} \frac{4^i}{2^i}$$

$$= 6 \cdot 4^n + 2^n \sum_{i=0}^{n-1} 2^i = 6 \cdot 4^n + 2^n \cdot \frac{1-2^n}{1-2} = 6 \cdot 4^n - 2^n(1-2^n)$$

estimate

$$\boxed{= 7 \cdot 4^n - 2^n}$$

$g(n) = \sum_{i=0}^{n-1} r^i = \frac{1-r^n}{1-r}$

geometric series

can factor out terms in summation without the counter (i)

$g(i) = i$

$3^{n-i} = \frac{3^n}{3^i} = 3^n\left(\frac{1}{3}\right)^i$

consecutive integers

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

(b) Again we apply the theorem above to obtain

$$T(n) = 10\cdot 3^n + \sum_{i=1}^{n} 3^{n-i}\cdot i = 10\cdot 3^n + 3^n \sum_{i=1}^{n}\frac{i}{3^i} = 10\cdot 3^n + 3^n \boxed{\sum_{i=1}^{n} i\cdot\left(\frac{1}{3}\right)^i}$$

To simplify the right-most summation above, we employ the following formula, valid for all $x \neq 1$:

let $x = \frac{1}{3}$

$$\boxed{\sum_{i=1}^{n} i\cdot x^i = \frac{nx^{n+2}-(n+1)x^{n+1}+x}{(1-x)^2}}$$

& $\sum_{i=1}^{n}\left(\frac{1}{3}\right)^i =$ formula for geometric series

It follows that

$$T(n) = 10\cdot 3^n + 3^n \sum_{i=1}^{n} i\cdot\left(\frac{1}{3}\right)^i = 10\cdot 3^n + 3^n\cdot\frac{n(1/3)^{n+2}-(n+1)(1/3)^{n+1}+1/3}{(1-1/3)^2}$$

$$= 10\cdot 3^n + 3^n\left(\frac{n}{4\cdot 3^n} - \frac{3(n+1)}{4\cdot 3^n} + \frac{3}{4}\right) = 10\cdot 3^n + \frac{n}{4} - \frac{3(n+1)}{4} + \frac{3}{4}\cdot 3^n$$

$$= \frac{43}{4}\cdot 3^n - \frac{n}{2} - \frac{3}{4}$$

[efficient]

efficient if $n = 100$ can be solved simply

$T(1) \leftrightarrow T(0)$

Let $n = 1$, then solve for $T(0)$

Ex. $T(1) = 4T(0) + 5$

$T(0) = \frac{T(1) - 5}{4}$

$= -\frac{3}{4}$

divide & conquer

restriction:

$T(n) = T\left(\frac{n}{2}\right) + 1$ when $n = 2$

$T(8) = T(4) + 1$

↳ relates term halfway back

involves powers of 2

$n = 2^k$

$T(n) = 2T(n-1)+1$
$T(5) = 2T(4)+1$] one term back to get next (related)

first order

4.3 Growth Rates of Solutions to Recurrences

some fraction of the way back

Divide and Conquer Algorithms

Definition: A *divide-and-conquer* recurrence for a sequence $T(n)$ is one of the form

$$T(n) = T(n/k) + C(n)$$

where k is constant and $C(n)$ is a given sequence.

single

Example: (Binary search) Consider a game where a person randomly picks a number x between 1 and 100 and you must guess what it is. Suppose you are allowed to ask one of two questions:

1. Is the number x greater than k?
2. Is the number x equal to k?

Determine an algorithm to find the number x and find a formula for $T(n)$, the number of questions needed to guarantee finding x between 1 and n. (See power point presentation)

each time you half the range, +1 question

Solution: We employ a binary search algorithm to find x inside the interval $I = [1, n]$.

1. If $|I| = 1$, then $x = I$. QUIT.
2. Else divide the interval $I = [1, n]$ into two equal sub-intervals: $I_1 = [1, n/2]$ and $I_2 = [n/2+1, 100]$. NOTE: If n is odd, then replace $n/2$ by $\lfloor n/2 \rfloor$.
3. Ask the question "Is the number x greater than 50?"
4. If yes, then replace I by I_2 and go back to Step 1.
5. If no, then replace I by I_1 and go back to Step 1.

To find a formula for $T(n)$, observe that finding a number say between 1 and 100 requires only one additional question ("Is the number greater than 50?") in comparison to finding a number between 1 and 50 (or between 51 and 100). Thus, we have the recurrence

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

Iterating this recurrence yields the first few values for $T(n)$ assuming $n = 2^k$ (power of 2):

$$T(1) = 1$$
$$T(2) = T(2/2) + 1 = T(1) + 1 = 1 + 1 = 2$$
$$T(4) = T(4/2) + 1 = T(2) + 1 = 2 + 1 = 3$$
$$T(8) = T(8/2) + 1 = T(4) + 1 = 3 + 1 = 4$$

Think of n in powers of 2

$k = \log n$ (base 2)

We infer from the pattern the following formula for $T(2^k)$:

$$T(2^k) = k + 1 \quad \text{(or equivalently, } T(n) = 1 + \log n\text{)}$$

$A = [7, 2, 4, 6, 1, 5, 8, 3]$

Example: (Merge sort) The following algorithm recursively sorts a list $A = \{i_1, i_2, \ldots, i_n\}$ that contains n items (we assume n is a power of 2):

1. Divide A into two equal disjoint subsets: $A_1 = \{i_1, \ldots, i_{n/2}\}$ and $A_2 = \{i_{n/2+1}, \ldots, i_n\}$
2. Sort A_1 and A_2 where each subset contains $n/2$ elements.
3. Merge A_1 and A_2.

$A = A_1 \cup A_2$
$A_1 = [7, 2, 4, 6]$
$A_2 = [1, 5, 8, 3]$

$A_1 = A_{11} \cup A_{12} = [7, 2] \cup [4, 6]$
$A_2 = A_{21} \cup A_{22} = [1, 5] \cup [8, 3]$

$T(n) =$ # comparisons needed to sort A
$=$ # comparisons to merge parts together (A_1 & A_2)

lists of size n you make n comparisons

Let $T(n)$ denote the number of steps needed to sort A (n items). Then the number of steps required to sort A_1 is the same as A_2, namely $T(n/2)$. Moreover, the number of steps needed to merge A_1 and A_2 equals n. Thus, we have the recurrence

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

To find an explicit formula for $T(n)$, we can iterate it to obtain the first few values:

$$T(1) = 1 \quad \text{(Why not 0?)}$$
$$T(2) = 2T(2/2) + 2 = 2T(1) + 2 = 2 \cdot 1 + 2 = 4 = 2 \cdot 2$$
$$T(4) = 2T(4/2) + 4 = 2T(2) + 4 = 2 \cdot 4 + 4 = 12 = 3 \cdot 4$$
$$T(8) = 2T(8/2) + 8 = 2T(4) + 4 = 2 \cdot 12 + 8 = 32 = 4 \cdot 8$$
$$T(16) = 2T(16/2) + 16 = 2T(8) + 16 = 2 \cdot 32 + 16 = 80 = 5 \cdot 16$$

factors to include n

$T(n) = 2$(work needed to sort each half) + work to merge

We infer from the pattern that

$k = \log n$ (base 2)

$$T(2^k) = (k+1) \cdot 2^k$$

or equivalently, if we set $n = 2^k$, then

$$T(n) = (\log_2 n + 1)n = n\log_2 n + n$$

$\to = O(n \log(n))$

Recursion Trees

Another method to determine an explicit formula for the divide-and-conquer recurrence

$$T(n) = \begin{cases} aT(n/p) + b & \text{if } n > 1 \\ T(1) & \text{if } n = 1 \end{cases}$$

$\to$ size of problem

is to iterate backwards and draw a recursion tree to keep track of the number of steps performed at each level of the recursion. Towards this end, we interpret the recurrence as follows:

- n – Size of problem
- $T(n)$ – Total amount of work needed to solve a particular problem of size n (binary search = # of questions)
- a – Number of sub-problems that the original problem is divided into
- n/p – Size of each sub-problem
- b – Amount of work needed to merge the sub-problems back to the original problem

We then compute the amount of work done at each level, referred to as the work per level, based on the formula

$$\boxed{\text{Work per level} = (\text{number of problems})(\text{work per problem})}$$

Lastly, we compute an explicit formula for $T(n)$ by summing up the work at all levels.

2 parts — size of each part

Example: Draw a recursion tree for the merge-sort recurrence:

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

work needed to merge (input size)

Assume n is a power of 2.

Solution: The recursion tree below describes the work done per level where the first row represents Level 0. Summing up the work from Level 0 to Level i corresponds to the i-th iteration of the recurrence:

tells you to split into 2 parts

$$T(n)=2T(n/2)+n$$
$$=2[2T(n/4)+n/2]+n=4T(n/4)+2n$$
$$=4[2T(n/8)+n/4]+2n=8T(n/8)+3n$$
$$\ldots$$
$$T(n)=2^iT(n/2^i)+in \qquad \text{(Level } i\text{)}$$
$$\ldots$$

merge sort

amount to merge

If we have $n=2^k$, then the recursion terminates at level $k=\log_2 n$. Substitution then gives the formula

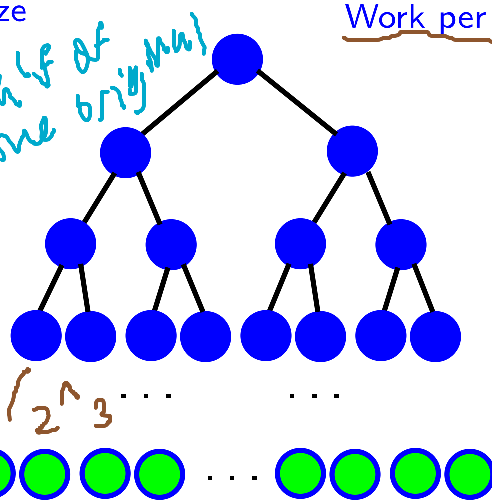
to see the pattern

$$T(n)=2^kT(n/2^k)+kn=nT(1)+kn=n[T(1)+\log_2 n]$$

On the other hand, we can draw a recursion tree for this recurrence:

# of levels = k+1

Lvl	# of Probl	Probl Size	Work per Probl	Work per Level
0	$1 \to 2^0$	n	n	$1 \cdot n$ n
1	$2 \to 2^1$	$n/2$	$n/2$	$2(n/2)=n$
2	$2^2=4$	$n/4$	$n/4$	$4(n/4)=n$
3	$2^3=8$	$n/8$	$n/8$	$8(n/8)=n$
		$\hookrightarrow n/2^3$		
k	$2^{\log n}=n$	1	$T(1)$	$nT(1)$



half of the original

sum of work @ all levels

stop when problem size 1

Since the work per level is same for all levels (except the last) and there are $(1+\log_2 n)$ levels, it follows that the total work is given by the same formula

$$T(n)=nT(1)+\overbrace{n+n+\ldots+n}^{k\text{ times}}=nT(1)+kn=n[T(1)+\log_2 n]=\Theta(n\log_2 n)$$

remember assumption $2^k=n$

not bigger than $O(n^2)$

Example: (See power point slides)

$T(n)=T(n/2)+1$
$T(n)=T(n/2)+n$
$T(n)=2T(n/2)+n$
$T(n)=4T(n/2)+n$

Three Different Behaviors

Lemma: Suppose $T(n)$ satisfies the recurrence
$$T(n)=aT(n/2)+n$$
where a is a positive integer and $T(1)$ is non-negative. Then we have the following big Θ bounds on $T(n)$:

1. If $a<2$, then $T(n)=\Theta(n)$.
2. If $a=2$, then $T(n)=\Theta(n\log n)$.
3. If $a>2$, then $T(n)=\Theta(n^{\log_2 a})$.

$3=\log_2 8 \longleftrightarrow 2^3=8$

$8T(n/2)+n$
$T(n)=\Theta\, n^3$

$T(n) = a \cdot \boxed{T(n-1)} + b$

Viewing Divide-and-Conquer Recurrences as Linear Recurrences

transform to first order linear

In general, a divide-and-conquer recurrence, say

$$T(n) = a\boxed{T(n/2)} + b,$$

can be viewed as a linear recurrence by making a change of variables $n = 2^k$,

$$T(2^k) = aT(2^{k-1}) + b,$$

and expressing the recurrence as a function of k, say by setting $S(k) := T(2^k)$:

$$S(k) = aS(k-1) + b$$