Data Structures and Algorithms

Spring 2009-2010

Outline

1. Mathematics Review (contd.)
2. Proof Techniques
   - Induction
   - Contradiction
   - Counterexample
   - Contrapositive
3. Recursion
   - Recursion Recap
   - Mergesort

Geometric Series

$$
\begin{aligned}
a + ar + ar^2 + ar^3 + \cdots + ar^{n-1} &= \frac{a(r^n - 1)}{r - 1} \\
&= \frac{a(1 - r^n)}{1 - r}
\end{aligned}
$$

Therefore

$$\sum_{i=0}^{n-1} 2^i = 1 + 2^1 + 2^2 + \cdots + 2^{n-1} = 2^n - 1$$

If $0 < r < 1$, then the n-term sum:

$$\sum_{i=0}^{n-1} r^i < \frac{1}{1 - r}$$

Geometric Series (contd.)

If $0 < r < 1$, then the infinite sum:

$$\sum_{i=0}^{\infty} r^i \approx \frac{1}{1-r}$$

A sum that often arises:

$$\sum_{i=1}^{\infty} \frac{i}{2^i}$$

NB: This is *not* the same as $\sum_{i=1}^{\infty} \frac{1}{2^i}$

Arithmetic Series

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

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \approx \frac{n^3}{3}$$

$$\sum_{i=1}^{n} i^k \approx \frac{n^{k+1}}{|k+1|}, k \neq -1$$

$$\sum_{i=1}^{n} \frac{1}{i} = H_n \approx \log_e n$$

Arithmetic Series (contd.)

$$\sum_{i=1}^{n} f(n) = n \times f(n) \qquad \text{Note: not } f(i)$$

$$\sum_{i=c}^{n} f(i) = \sum_{i=1}^{n} f(i) - \sum_{i=1}^{c-1} f(i)$$

Four Main Proof Techniques

When trying to formally establish some claim, the main proof techniques that we use are:

- *induction*
- *contradiction*
- *counterexample*
- *contrapositive*

Outline

Proof by Induction

Suppose we have a theorem quantified by a variable n

- proof by induction proves that the theorem is true for $n = K + 1$ by using the fact that the theorem is true for some number K
- only works for proving things about integers or other “discrete” objects
- in the inductive step, we show that the theorem is true for $n = K + 1$ assuming that it is true for $n = K$
- for this to work we also need a starting point or, *base* case

Prove

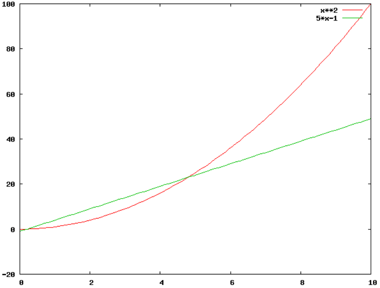
$$\sum_{i=1}^{n}(2i - 1) = n^2, \quad n > 0$$

Proof by Induction (contd.)

Prove

$$5n - 1 < n^2, n \geq 5$$

Warning: This doesn't hold for when n is "small" ($n \leq 4$), so we have to build this in to our theorem



Red, curved line is a plot of n^2; green, straight line is $5n - 1$

Outline

Proof by Contradiction

- Proof by contradiction relies on fact that everything either *is* or *isn't* true
- To prove some fact, we attempt to prove the **opposite** hypothesis instead
- By encountering a contradiction somewhere in this "proof" it shows that our new hypothesis must be false, meaning that the original was true
- So to prove the theorem "$A \Rightarrow B$", we would attempt to prove the *opposite*, "$A \Rightarrow \neg B$"

Prove The sky over Limerick is sometimes grey.

Proof: Try to prove opposite: "The sky over Limerick is *never* grey." Look outside now. It's grey! But this contradicts our "never grey" statement. So the original hypothesis ("sometimes grey") must be true

Proof by Contradiction (contd.)

Prove

Prove there are an infinite number of prime numbers.

Proof: Suppose this is not true. Then we *assume* there are only a finite number of primes, say, m of them. Then we can write these m numbers down:

$$p_1, p_2, p_3, \ldots, p_m$$

where p_m is the largest prime possible. (Background fact: every **non-prime** number can be broken into its "factors"; this can be done to each of these factors (repeatedly) until you have only prime numbers.) Now consider the number $K = 1 + p_1 . p_2 . p_3 . \cdots . p_m$. This number cannot have any divisor (a number that divides it cleanly) so it must be prime. But $K > p_m$, which contradicts our assumption. Each step in this proof is sound, so our modified hypothesis is unsound.

Outline

1. Mathematics Review (contd.)
2. Proof Techniques
   - Induction
   - Contradiction
   - Counterexample
   - Contrapositive
3. Recursion
   - Recursion Recap
   - Mergesort

(Dis)Proof by Counterexample

Can only use this to prove a statement false.
Find an instance where the theorem / claim is false.

Prove

$$5n + 1 < n^2, \quad n > 0$$

Proof: When $n = 1$, $5 + 1 \not< 1$.

Prove "The sun always shines in Limerick."

Proof: Take a look outside: the sun is *not* shining!

Outline

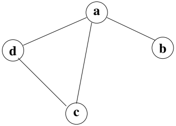
1. Mathematics Review (contd.)
2. Proof Techniques
   - Induction
   - Contradiction
   - Counterexample
   - Contrapositive
3. Recursion
   - Recursion Recap
   - Mergesort

Proof by Contrapositive

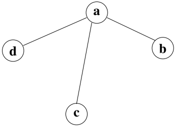
To prove "$A \Rightarrow B$", prove instead "$\neg B \Rightarrow \neg A$"
Both are equivalent; technique is often called *Modus Tolens*

From CS4115 Final Exam, 2001-2002

"Prove that if every vertex in a graph $G = (V, E)$ has degree greater than 1, then there must be some cycle in the graph."



A Graph



A Tree

Note carefully what the theorem says: if every vertex has degree of two, or more, then there must be a cycle, yet even if some vertices have degree 1, there could be still a cycle (pic. on left).

Proof by Contrapositive (contd.)

Proof (1): (Follow your nose.)

- If every vertex has degree 2 or more, then it has an in-edge and an out-edge
- Enter the vertex, v, on one and exit the vertex on one of the other edges; “mark” the vertex before exiting
- This takes us to a new vertex and, since its degree is greater than 1, we can exit *it* on a different edge, also

Proof by Contrapositive (contd.)

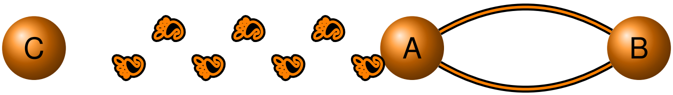
- Keep doing this, marking each vertex as we go, and never walking across an edge twice
- Since there are a finite number of vertices, we *must* visit some vertex that we've seen before
- This is the cycle!

The trouble with walking to vertices of degree 1 (C below):

Proof by Contrapositive (contd.)

- Keep doing this, marking each vertex as we go, and never walking across an edge twice
- Since there are a finite number of vertices, we *must* visit some vertex that we've seen before
- This is the cycle!

The trouble with walking to vertices of degree 1 (*C* below):



Proof by Contrapositive (contd.)

Proof (2): (Use every trick in the book.)
If we abbreviate "there must be some cycle in the graph" by "$cyc(G)$" (G has a cycle) then we can write the theorem as:

$$\forall v \in V, d(v) \geq 2 \quad \Rightarrow \quad cyc(G)$$

The contrapositive of this is "if there is no cycle in G then there must be some vertex v whose degree is not 2 or more", which can be written

$$\neg cyc(G) \quad \Rightarrow \quad \exists v \in V, \neg d(v) \geq 2$$

$$\neg cyc(G) \quad \Rightarrow \quad \exists v \in V, d(v) \leq 1$$

Proof by Contrapositive (contd.)

To prove this, we will look for a *contradiction* in proving the opposite

$$\neg cyc(G) \quad \Rightarrow \quad \forall v \in V, d(v) \geq 2$$

This cannot be true because the previous tree clearly has no cycle yet the leaves have degree 1.
Thus, the following is false

$$\neg cyc(G) \quad \Rightarrow \quad \forall v \in V, d(v) \geq 2$$

meaning, both

$$\neg cyc(G) \quad \Rightarrow \quad \exists v \in V, \neg d(v) \geq 2$$

and

$$\forall v \in V, d(v) \geq 2 \quad \Rightarrow \quad cyc(G)$$

are true.

Outline

Recursion Recap

- To solve a problem *recursively* we break the problem in to one or more subproblems that have the same description as the original
- Each of the subproblems can then be solved using the algorithm
- There will likely be other housekeeping to do as well as part of the recursive call

Outline

Mergesort Requires Recursion

To sort an array, split the array in two, sort the two halves (independently) and then merge the two halves in to one.

```
void mergesort(int arr[], int lo, int hi)
{ // sort array arr, from lo to hi

   if (lo >= hi) return;    // base case

   int mid = (lo + hi) / 2;

   mergesort(arr, lo, mid);
   mergesort(arr, mid+1, hi);

   merge_halves(arr, lo, mid, hi); // housekeeping
}
```

The Merge in Mergesort

```
// merge_halves merges sorted halves arr[lo..mid]
//    and arr[mid+1..hi] to give the sorted arr[lo.
void
merge_halves(int arr[], int lo, int mid, int hi)
{
  // homework exercise
}
```