Lecture 12: 5/12/2009

Announcements: Ps7 out today. Ps6, Midterm solutions out soon. Midterm info on Web page.
Midterm back, 12

Asymptotics (review)

O(g) = {f: \N -> \R: \exists C, N s.t.
f(n) <= C g(n) for all n>= N}

Θ(g) = {f: \N -> \R: \exists c,C N s.t.
c g(n) <= f(n) <= C g(n) for all n>= N}

Example:
True/False:

If f is Θ(n^2) then f is O(n^2) TRUE
n! = O(2^n) NO
n! = O(n^n) YES
(Truth: n! = Θ((n/e)^n sqrt(n)))

Claim: H_n = 1/1 + 1/2 + ... + 1/n = O(lg n)
Draw picture.

Upperbound by 1 + integral_1^n (1/x)dx = 1 + ln(n) = O(lg n)
show Could I write O(log n)
Sure; O(log n) = O(lg n)

(following not done in class): Compute the asymptotic running time of the following algorithm:
Given: a formula phi, decide if phi is satisfiable
by trying all possible truth assignments.
Suppose phi has n variables and takes m bits to write down.

Answer: Need to try 2^n t.a. How long to try teach? O(m).
All together? O(m 2^n).

One reason asymptotic notation handy:

O(n^2) + O(n^2) = O(n^2)
O(n^2) + O(n^3) = O(n^3)
O(n log n) + O(n) = O(n log n)
etc.

A reason we use asymptotic notation: model independence

How long does the following code take to run, for searching for x in an array A:

```
  for i=1 to n do
       (If (A[i]=x) Return(i);
Return "not found"
```

Takes at most c1*n + c2 where c1 is the work to do one iteration of the for loop, and c2 is the max time for the final return. The total time is O(n).

Faster algorithms can do this in O(logn) time or even O(1) if we can preprocess the array A.

How long to sort by selection sort? (repeatedly find the smallest remaining element)
O(n^2)
Already enough to distinguish from more sophisticated algorithms.

----
2. How big is that infinity? (3.7 in book)

Puzzle: Which are there more of: natural number or integers?
Mathematicians answer: they are the same. Not because they are both infinite, but because there is a bijective function
f: *N* -> \Z

Recall

e.g,
f(x) = \lceil x/2 \rceil if x is odd
-x/2 +1 if x is even

Is there a bijective function from \Z to *N*

Answer 1: yes, automatic from above
Answer 2, yes, f(x) = -2x if x even
2x-1 if x is odd

Puzzle: is there a bijective function from *N* to *N* x *N (or equivalently to Q+).*
Technique: "Dovetailing" // Consider a 2-D matrix with the numbers 1,2, … on the top and right axis. Label the [1,1] entry 1, then the [1,2] entry 2 and the [2,1] entry 3 (going top to bottom along the diagonal going from right to left). Then for the next diagonal give [1,3], [2,2], [3,1] 4,5,6 respectively. Continue on each diagonal …

Definition: Let A and B be sets. We say that A is *equinumerous* to B,
A ~ B, if there is a bijection f from A to B.

Definitions: (not all of these below were done in class)
A set is *finite* if it is equinumerous to I_n = {1,...,n} for some n ∈ *N*.
A set is *infinite* if it is not finite.
A set is *countably infinite* it is equinumerous to ***N***
A set is *countable* if it is finite or countably infinite.
A set is *uncountable* if it is not countable.

Proposition: ~ is an equivalence relation

Proposition: *Q* is countably infinite.

Proposition: if A and B are countable then so is A x B and A u B

Theorem: P*(N*) is uncountable. (the powerset of the natural numbers)

Proof. Suppose for contradiction that P(*N*) were countable:
say P(***N***) = {A_1, A_2, ...}
We construct a subset B of ***N*** as follows
B contains i∈ iff i is not in A_i
Now B is a subset of *N* so it must be in the enumeration: say B = A_j

But: is j ∈ B?
   j ∈ B iff j\not∈ A_j = B
Contradiction.

Definition: For sets A, B, write A \le B if there exists a one-to-one function
  f: A -> B.

Theorem: [Cantor-Schr\"oder-Bernstein]   [not done in class]
If A \le B and B \le A then A ~ B