Mathematical Foundations of Computer Science

Ten Powerful Ideas

and applications

Final Exam Solutions	Bruce Maggs	May 4, 1998

Repeat After Me.

This part is to test your ability to study material and repeat it.

1. [3 points] **Contrapositive.**
What is the contrapositive of the assertion "A implies B"?

not B implies not A

2. [3 points] **Geometric series.**
Give a closed-form expression for $\sum_{i=0}^{+\infty} x^i$, *where* $|x| < 1$.

$$\frac{1}{1-x}$$

3. [3 points] **Group size.**
State Lagrange's Theorem.

For any group G and subgroup H of G, $|H|$ divides $|G|$.

4. [3 points] **Polynomial interpolation.**
In the finite field GF_p, *how many degree-*$(n-1)$ *polynomials pass through the* k *points* $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, ($
(Assume $k \leq n$.*)*

Since n points determine a degree-$(n-1)$ polynomial, you can choose the remaining $n-k$ points arbitrarily. Each point has p choices, so there are p^{n-k} polynomials. (Note that this answer also assumes that $n \leq p$; otherwise you won't be able to find n distinct points x_i to determine the polynomial.)

5. [3 points] **One-to-one function.**
Let f *be a function from a set* A *to a set* B. *Give a necessary and sufficient condition for* f *to be a one-to-one function.*

For all x and y such that $x \neq y$, $f(x) \neq f(y)$.

6. [3 points] **The halting problem.**
Give an example of a real number that is describable but not computable.

The real number between 0 and 1 whose decimal representation is defined as follows: the ith digit after the decimal point is 1 if the ith computer program halts when run on a description of itself, and 0 otherwise. If a program existed to compute the digits of this number, then you could use it to solve the halting problem.

7. [3 points] **Finite probability space.**
Suppose that Ω is a finite set of outcomes and that p is a function from subsets of Ω (called events) to real numbers in the range $[0,1]$. What two conditions must p satisfy in order for Ω and p together to form a probability space?

1. $p(\Omega) = 1$
2. For all $E \subseteq \Omega$,
$$p(E) = \sum_{\omega \in E} p(\omega)$$

8. [3 points] **Fermat's Little Theorem.**
State Fermat's Little Theorem.

For any prime p and $1 \leq a \leq p-1$,

$$a^{p-1} \equiv 1 \pmod{p}$$

9. [3 points] **Big-Omega notation.**
Suppose that f and g are both functions from the natural numbers to the natural numbers. Give a precise definition of the notation $f(n) = \Omega(g(n))$.

There exists $c > 0$ and $N \geq 0$ such that for all $n \geq N$,

$$f(n) \geq cg(n)$$

10. [3 points] **Field.**
The following definition of a field has one mistake in it. What is the mistake? A set S together with an addition operator $\oplus$ and a multiplication operator $\odot$ form a field if S and $\oplus$ form a commuative group, S and $\odot$ form a commutative group, and $\odot$ distributes over $\oplus$.

$S - 0$ and $\odot$ form a commutative group (not S and $\odot$), where 0 is the identity element of $\oplus$.

11. [5 points] **Horner's method.**
Suppose that the n coefficients of a degree $n-1$ polynomial $p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_0$ are stored in an array a, so that the coefficient of x^i in $p(x)$ is $a[i]$. Write a piece of C or C++ code to evaluate p at some point $x = b$, i.e., evaluate $p(b)$. Ignoring calculations of loop indices, your program must perform at most $n-1$ additions and $n-1$ multiplications.

Ignoring variable declarations,

```
p = a[n-1];
for (i=n-2; i >= 0; i--)
    p = x*p + a[i];
return p;
```

12. [5 points] **Euclid's algorithm.**
Given two non-negative integers a and b, where $b \geq a$, Euclid's algorithm finds the greatest common divisor of a and b. Give a short recursive description of Euclid's algorithm (in pseudocode or C/C++).

```
gcd (a, b)
    if a = 0 then return b
    else return gcd (b mod a, a)
```

Basic Techniques.

This part will test your ability to reason with some of the basic techniques learned in the class.

13. [8 points] **Recurring nightmares.**
Give a closed-form solution to the following recurrence. Show your work.

$$\begin{aligned} f(n) &= 5f(n-2) - 4f(n-4) \\ f(0) &= 2 \\ f(1) &= 0 \\ f(2) &= 8 \\ f(3) &= 0 \end{aligned}$$

Try $f(n) = \lambda^n$:

$$\lambda^n = 5\lambda^{n-2} - 4\lambda^{n-4}$$

$$\begin{aligned} \lambda^{n-4}(\lambda^4 - 5\lambda^2 + 4) &= 0 \\ \lambda^{n-4}(\lambda^2 - 4)(\lambda^2 - 1) &= 0 \\ \lambda^{n-4}(\lambda - 2)(\lambda + 2)(\lambda - 1)(\lambda + 1) &= 0 \end{aligned}$$

So the possible solutions are $2^n, (-2)^n, 1^n, (-1)^n$.
Try $f(n) = a2^n + b(-2)^n + c1^n + d(-1)^n$:

$$\begin{aligned} f(0) = a + b + c + d &= 2 \\ f(1) = 2a - 2b + c - d &= 0 \\ f(2) = 4a + 4b + c + d &= 8 \\ f(3) = 8a - 8b + c - d &= 0 \end{aligned}$$

Solving for the unknowns gives:

$$\begin{aligned} a &= 1 \\ b &= 1 \\ c &= 0 \\ d &= 0 \end{aligned}$$

So the answer is

$$f(n) = 2^n + (-2)^n$$

14. [8 points] **Seven-card stud.**
Seven-card stud is a variation of poker in which each player is dealt seven cards, and then selects any five of the seven to form a poker hand.

How many times more likely is it to receive a hand containing four-of-a-kind in seven-card stud than in the version of poker in which each player is dealt five cards and cannot discard or exchange any cards? (We're looking for the ratio of the probabilities.)

$$\begin{aligned} \Pr(\text{4 of a kind in 5-card stud}) &= \frac{13 \cdot 48}{\binom{52}{5}} \\ \Pr(\text{4 of a kind in 7-card stud}) &= \frac{13 \cdot \binom{48}{3}}{\binom{52}{7}} \end{aligned}$$

$$\begin{aligned} \frac{\Pr(\text{4 of a kind in 7-card stud})}{\Pr(\text{4 of a kind in 5-card stud})} &= \frac{\frac{13 \cdot \binom{48}{3}}{\binom{52}{7}}}{\frac{13 \cdot 48}{\binom{52}{5}}} \\ &= \frac{\binom{48}{3}\binom{52}{5}}{48\binom{52}{7}} \\ &= \frac{\frac{48 \cdot 47 \cdot 46}{3 \cdot 2}\frac{52 \cdot 51 \cdot 50 \cdot 49 \cdot 48}{5 \cdot 4 \cdot 3 \cdot 2}}{48\frac{52 \cdot 51 \cdot 50 \cdot 49 \cdot 48 \cdot 47 \cdot 46}{7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2}} \\ &= 7 \end{aligned}$$

15. [8 points] **Even worse than taking this final.**
Four graders have 112 final exams to grade.

1. *(4 points) How many different ways are there of dividing up the 112 exams among the 4 graders? (We don't care if some graders get more work than others.)*

   Each exam can be assigned to any grader independently,

$$4^{112} \text{ ways}$$

2. *(4 points) Suppose that the graders are not concerned with the names of the students on the exams that they are to grade, but only about the number of exams that they are to grade. How many different ways are there of partitioning the exams when they are all indistinguishable (again, not caring if some graders get more work than others)?*

   Rearrange the 112 identical exams with $4 - 1 = 3$ identical separators,

$$\binom{115}{3} \text{ ways}$$

A Moment's Thought!

This section tests your ability to think a little bit more insightfully. You must give complete explanations of your answers. **Do three of the five problems. Do not do all five; if you do we will grade three of our choice.**

16. [12 points] **We really do think like this.**
Prof. Baraff and Prof. Maggs are competing to see who can give harder homework assignments. For each assignment, Prof. Maggs wins with probability 1/2 and Prof. Baraff wins with probability 1/2, independent of the outcomes of the previous assignments. The competition ends when one of these (admittedly somewhat evil) professors has won on two more of the assignments than the other one. At this point the professors stop giving out assignments. How many 15-299 assignments do you expect to have to do?

Hint: Look at two assignments at a time.

Consider the possible outcomes of each pair of assignments (where B represents a win by Baraff and M represents a win by Maggs): BB, MM, BM, and MB, each with probability 1/4. As long as BM or MB occur (probability 1/2), the score is tied and the game keeps going. As soon as BB or MM occurs (probability 1/2), the game is over. The expected number of pairs of assignments needed for a BB or MM to appear is 2, so the expected number of assignments is 4.

17. [12 points] **Defying all expectations.**
Provide a counterexample to the following hypothesis.

Hypothesis. For every finite probability space and every random variable x defined on that probability space, for any function f mapping the reals to the reals,

$$E[f(x)] = f(E[x]).$$

Give a counterexample (a random variable x and a function f that falsify the hypothesis) and show why.

Let $f(x) = 2^x$ and x be the random variable which is 0 with probability 1/2 and 1 with probability 1/2. Then

$$\begin{aligned} E[f(x)] &= \frac{1}{2} \cdot 2^0 + \frac{1}{2} \cdot 2^1 = \frac{3}{2} \\ f(E[x]) &= f(\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1) = f(\frac{1}{2}) = 2^{\frac{1}{2}} = \sqrt{2} \end{aligned}$$

Thus $E[f(x)] \neq f(E[x])$.

18. [12 points] **Sign on the dotted line.**
In the RSA public-key cryptography system, a plaintext message m from Bob to Alice is encrypted using Alice's public key $\langle e, n \rangle$ by computing $m^e \bmod n$. Alice, who has a private key d such that $e \cdot d = 1 \bmod \phi(n)$, then decrypts the message by computing $(m^e)^d = m^{ed} = m \bmod n$.

One drawback to the scheme, as described so far, is that there is no way for Alice to tell if the message really was sent by Bob, or if it was forged by an imposter. To remedy this problem, a signature scheme is used. Suppose that Bob's public key is $\langle e', n' \rangle$, and his private key is d', such that $e' \cdot d' = 1 \bmod \phi(n')$. Rather than sending m^e to Alice, Bob sends $(m^{d'} \bmod n')^e \bmod n$.

1. *(4 points) Explain how Alice recovers the original plaintext message m.*

   She applies her private key d to get $((m^{d'} \bmod n')^e)^d \bmod n = m^{d'} \bmod n'$. Then she applies Bob's public key e' to get $(m^{d'})^{e'} \bmod n' = m$.

2. *(4 points) Why can Bob be confident that revealing both m and $m^{d'} \bmod n'$ to Alice is a safe thing to do?* Hint: How can Alice generate a similar pair of numbers x and $x^{d'} \bmod n'$ by herself, using only public information?

   Bob might be worried that revealing m and $m^{d'} \bmod n'$ to Alice might give her the ability to figure out his private key d'. Here's why she can't. Suppose Alice had a procedure that used the pair x and $x^{d'} \bmod n'$ to compute d'. Then she could take any number y and compute $y^{e'} \bmod n'$ to form a pair suitable for this procedure (where $x = y^{e'} \bmod n'$ and $x^{d'} = (y^{e'})^{d'} \bmod n' = y$). So if she had such a procedure, she would be able to get Bob's private key by herself! Bob isn't revealing anything new.

3. *(4 points) Why is Alice confident that the message $(m^{d'}) \bmod n'$ was encrypted by Bob and not by someone else?*

   Only someone who knows d' can compute $(m^{d'}) \bmod n'$. Since only Bob knows d', the message must have been encrypted by Bob.

19. [12 points] **That's irrational.**
A real number is called rational if it can be expressed as the ratio of two integers, and irrational otherwise. In solving this problem, you may assume that $\sqrt{2}$ is irrational. (It is.)

1. *(4 points) Prove that if x is a rational number, then $x \cdot \sqrt{2}$ is an irrational number.*

This statement is actually true only if you exclude $x = 0$, so we'll do that in our proof. Proof by contradiction. Suppose for some rational $\frac{p}{q}$, $\frac{p}{q} \cdot \sqrt{2}$ is a rational number $\frac{p'}{q'}$. Then $\sqrt{2} = \frac{p'q}{q'p}$, so $\sqrt{2}$ is rational. This is a contradiction, so $\frac{p}{q} \cdot \sqrt{2}$ must be irrational for all $\frac{p}{q}$.

2. *(4 points) Let Q denote the set of rational numbers, and let T denote the set of numbers that are rational multiples of $\sqrt{2}$, i.e., $T = \{y : y = x \cdot \sqrt{2}, x \in Q\}$. Prove that there is a one-to-one and onto mapping from $Q \cup T$ to T.*

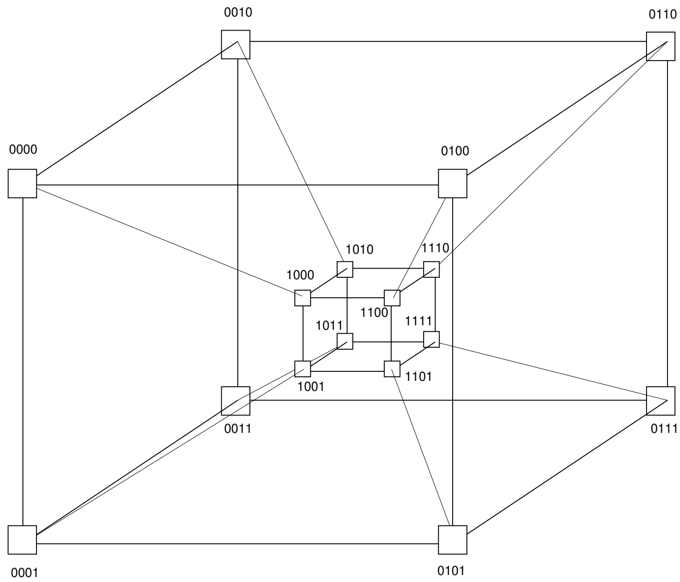
   The rational numbers Q are countable. T is also countable, because $f(x) = x \cdot \sqrt{2}$ is a one-to-one onto mapping from Q to T. So we can enumerate both Q and T. We can enumerate $Q \cup T$ by interleaving the enumerations for Q and T (including 0 only once, however), so $Q \cup T$ is countable. Since both T and $Q \cup T$ are countable, there must be a one-to-one onto mapping from $Q \cup T$ to T.

3. *(4 points) Prove that the set of real numbers has the same cardinality as the set of irrational numbers.* Hint: divide the real numbers into $Q \cup T$ and $\mathcal{R} - (Q \cup T)$.

   Form a one-to-one onto mapping from the reals to the irrationals as follows. Map elements in $Q \cup T$ (the rationals, and all irrationals that are rational multiples of $\sqrt{2}$) to their corresponding elements in T according to the one-to-one onto mapping from part (b). Map elements in $\mathcal{R} - (Q \cup T)$ (the irrationals that are *not* rational multiples of $\sqrt{2}$) to themselves. Putting these two one-to-one onto functions together covers all the irrational numbers. Thus the reals have the same cardinality as the irrationals.

20. [12 points] **Hypercubic.**

The hypercube of dimension k, abbreviated H_k, is a graph that can be constructed recursively by making two copies of H_{k-1} and joining corresponding nodes in the two copies by an edge. For the base case, H_0 is a single node. The hypercube H_4 is shown below.



1. *(6 points) Prove by induction that H_k has $k2^{k-1}$ edges.*

   First we need a lemma: the number of nodes in H_k is 2^k. By induction on k: H_0 has $2^0 = 1$ node. If H_{k-1} has 2^{k-1} nodes, then since H_k consists of two copies of H_{k-1} with no other nodes added, H_k has $2 \cdot 2^{k-1} = 2^k$ nodes.

   Now to prove that H_k has $k2^{k-1}$ edges, again by induction on k. For the base case, observe that H_0 has $0\dot{2}^{0-1} = 0$ edges. Now suppose H_{k-1} has $(k-1)2^{k-2}$ edges. The hypercube H_k consists of two copies of H_{k-1} with edges added between corresponding nodes, so H_k has $2 \cdot (k-1)2^{k-2} + 2^{k-1} = (k-1)2^{k-1} + 2^{k-1} = k2^{k-1}$ edges, as desired.

2. *(6 points) Prove that for all $k \geq 2$, H_k has a Hamiltonian tour. (Recall that a Hamiltonian tour is a simple cycle that visits all of the nodes in a graph.)*

   By induction on k, as before. The base case is H_2 (four nodes arranged in a square), which has a trivial Hamiltonian tour.

   Now suppose H_{k-1} has a Hamiltonian tour. The hypercube H_k consists of two copies of H_{k-1} with corresponding vertices joined. Let us denote these H_{k-1} subgraphs by H_{k-1} and H'_{k-1}. To make a Hamiltonian tour of H_k, start at any node x in H_{k-1} and follow the Hamiltonian tour of H_{k-1}. At the last node y of the tour, just before returning to x, move to the corresponding node y' in the other subgraph H'_{k-1}. Follow the Hamiltonian tour of H'_{k-1} backwards through corresponding nodes to reach x', then move back to its corresponding node x, completing the tour. Because the Hamiltonian tours of H_{k-1} and H'_{k-1} visit every node in their respective subgraphs exactly once, the overall tour visits every node in H_k exactly once.