Massachusetts Institute of Technology
6.042J/18.062J: Mathematics for Computer Science
Professors David Karger and Nancy Lynch

Handout 25
March 30, 2000

Problem Set 6 Solutions

Problems:

Problem 1 Determine the flaw in the following reasoning and explain your answer.

$$\left(\sum_{j=1}^{n} a_j\right)\left(\sum_{k=1}^{n} 1/a_k\right) = \sum_{j=1}^{n}\sum_{k=1}^{n} a_j/a_k = \sum_{k=1}^{n}\sum_{k=1}^{n} a_k/a_k = \sum_{k=1}^{n}\sum_{k=1}^{n} 1 = \sum_{k=1}^{n} n = n^2$$

Solution.

Here, the problem is taking the second step. It is incorrect to change the variable of the first summation to k since the second summation uses that variable. By doing this change of variables, we change the meaning of the double summation.

□

Problem 2 Find a closed-form expression for the summation $\sum_{i=1}^{n} i^2 x^i$.

Solution.

In the lecture and the lecture notes we derived the following:

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

Differentiating (with respect to x) the above expression and multiplying with x we find that

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

□

Problem 3 Show that $\sum_{k=1}^{\infty} 1/k^{3/2}$ is bounded from above by a constant.

Solution.

We use the integral method for bounding the infinite sum. Since $f(k) = \frac{1}{k^{3/2}}$ is monotonically decreasing for $k > 1$, we have:

$$\begin{aligned}
\sum_{k=1}^{\infty} 1/k^{3/2} &= 1 + \sum_{k=2}^{\infty} 1/k^{3/2} \\
&\leq 1 + \int_1^{\infty} \frac{1}{x^{3/2}} dx \\
&= 1 + \left[\frac{-2}{\sqrt{x}}\right]_1^{\infty} \\
&= 1 + (0 + 2) \\
&= 3
\end{aligned}$$

Hence, $\sum_{k=1}^{\infty} 1/k^{3/2} \leq 3$.

□

Problem 4 In this problem, we will prove that the "average" number of divisors of a number of size about n is "about" $\ln n$. More formally, let $t(n)$ be the number of divisors of the number n, and let

$$T(n) = \frac{1}{n} \sum_{j=1}^{n} t(j)$$

be the average number of divisors of all numbers less than or equal to n. Show that $T(n) \sim \ln n$.

Hint: Let the function $f(i, j) = 1$ if i is a divisor of j, and 0 otherwise. What do the row and column sums of $f(i, j)$ mean?

Solution.

Consider the image of the function $f(i, j)$ as a matrix whose rows and columns correspond to values of i and j respectively; that is, element (i, j) of the matrix denotes whether or not i divides j.

				j				
	$f(i,j)$	1	2	3	4	5	6	
	1	1	1	1	1	1	1	
i	2	0	1	0	1	0	1	$\cdots$
	3	0	0	1	0	0	1	
	4	0	0	0	1	0	0	
	$\vdots$			$\vdots$				

Notice that the sum of each column j corresponds to the number of divisors of the number j and the sum of each row i corresponds to the number of multiples of the number i. In fact, the sum of the first n elements of a row i correspond to the number of multiples of i up to the number n, which can be compactly represented as $\lfloor \frac{n}{i} \rfloor$.

It follows that:

$$\begin{aligned} T(n) &= \frac{1}{n}\sum_{j=1}^{n} t(j) \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} f(i,j) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left\lfloor \frac{n}{i} \right\rfloor \end{aligned}$$

$T(n)$ can then be bounded from below and above as follows:

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{n}{i}-1\right) &\le T(n) \le \frac{1}{n}\sum_{i=1}^{n}\frac{n}{i} \\ \frac{1}{n}\left(\sum_{i=1}^{n}\frac{n}{i}-\sum_{i=1}^{n}1\right) &\le T(n) \le \sum_{i=1}^{n}\frac{1}{i} \\ \sum_{i=1}^{n}\frac{1}{i}-1 &\le T(n) \le \sum_{i=1}^{n}\frac{1}{i} \end{aligned}$$

In class, we determined that $H_n = \sum_{i=1}^{n}\frac{1}{i} \sim \ln n$. Thus, it follows that $T(n) \sim \ln n$.

□

Problem 5 Asymptotic notation

For each pair of functions $(f(n), g(n))$ in the table below, indicate whether the statement at the top of the column is true (e.g., whether $f(n) = O(g(n))$). Assume $k \ge 1$, $\epsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of a table with "yes" or "no" in each box.

$f(n)$	$g(n)$	$f(n) = O(g(n))$	$g(n) = O(f(n))$	$f(n) = \Theta(g(n))$
$\log^k n$	n^ϵ			
n^k	c^n			
$\sqrt{n}$	$n^{\sin n}$			
2^n	$2^{n/2}$			
$\log(n!)$	$\log(n^n)$			

Solution.

$f(n)$	$g(n)$	$f(n) = O(g(n))$	$g(n) = O(f(n))$	$f(n) = \Theta(g(n))$
$\log^k n$	n^ϵ	yes	no	no
n^k	c^n	yes	no	no
$\sqrt{n}$	$n^{\sin n}$	no	no	no
2^n	$2^{n/2}$	no	yes	no
$\log(n!)$	$\log(n^n)$	yes	yes	yes

Following are some hints on deriving the table above:
(a) Refer to the lecture notes on O-notation.
(b) Polynomial growth versus exponential growth. (See Theorem 3.6, Lecture 9, Fall'97.)
(c) When n is a multiple of π, $n^{\sin n} = 1$, so no constant times 1 will bound $\sqrt{n}$ as n ranges over multiples of π. When n is of the form $2k'\pi + \pi/2$, for some integer k', then $n^{\sin n} = n$ and $\frac{n}{\sqrt{n}} = \sqrt{n}$ grows without bound as k' increases.
(d) $\frac{2^n}{2^{n/2}} = 2^{n/2}$ grows without bound as n grows — it is not bounded by a constant.
(e) See Rosen §1.8 Ex. 61, 62.

Problem 6 Recurrences

Consider a set of record items, each carrying some data and a key and the following definition of a *majority key.*

Definition. *The* majority key *of a set of m records is a key that occurs more than $m/2$ times.*

For example, 2 is the majority key in a set of 3 records which are tagged, consecutively, by keys 1, 2, and 2. On the other hand, in the set of 4 records tagged by keys 1, 2, 3, and 3, there is no majority key.

Consider the following algorithm for finding the majority key of a set of records: If there is an even number of records, we pair adjacent items up (first with second, third with fourth, etc.) and compare their keys. If they have the same key, we keep one of them, otherwise we throw them both away. We run this algorithm recursively on the resulting set. If there is an odd number of items in our set, we find the majority key of the set containing all but the last item. If there was no majority key then we check if the key of the last record is the majority key by explicitly counting how many times this key appears in the whole set. We continue in this fashion until we either get one record, in which case we check if its key is in fact the majority key — again by counting how many times this key appears in the original set — or we end up with no items, in which case we conclude that there was no majority key.

(a) Prove that if there is a majority key in the initial set of records then this algorithm terminates with a record item carrying that key. (Hint: Find an invariant that is preserved at every stage. Consider the even and odd cases separately.)

Solution.

We model the majority key algorithm as a state machine. The state of the state machine is defined as the tuple (a, b, s, r), where a corresponds to the initial list of records, b corresponds to the current list of records in which we are still looking for a majority key, s corresponds to a list of the odd records that we still need to test, and r corresponds to a list containing the majority key of the initial list of records a. The initial state of the state machine is the state $(a, a, \emptyset, \emptyset)$. The set of transitions $(a, b, s, r) \to (a', b', s', r')$ is described by the following pseudocode:

$\delta : (a, b, s, r) \rightarrow (a', b', s', r')$
  Precondition:
    $(b \neq \emptyset \vee s \neq \emptyset) \wedge r = \emptyset)$
  Effect:
    $a' = a$
    if $|b| > 1$ **then**
      if $|b|$ is even **then**
        $b' = \textit{pair-and-remove}(b)$
        $s' = s$
        $r' = r$
      else
        $b' = \textit{pair-and-remove}(b - \textit{last}(b))$
        $s' = \textit{append}(s, \textit{last}(b))$
        $r' = r$
    else
      if $|b| = 1$ **then**
        if $\textit{is-majority}(a, \textit{last}(b))$ **then**
          $b' = \emptyset$
          $s' = \emptyset$
          $r' = \{\textit{last}(b)\}$
        else
          $b' = \emptyset$
          $s' = s$
          $r' = r$
      else
        $b' = b$
        if $\textit{is-majority}(a, \textit{last}(s))$ **then**
          $s' = \emptyset$
          $r' = \{\textit{last}(s)\}$
        else
          $s' = \textit{remove-last}(s)$
          $r' = r$

The state machine thus terminates either when the majority key is stored in the list r, or when the current list of records b is empty and there are no more records in s to test. If upon termination r is empty, then the algorithm has determined that there is no majority key.

Let the predicate P on the states of the state machine be defined as follows: if x is a majority key of the initial list of records a, then x is either the majority element in b or is in $s \cup r$. According to the invariant theorem, to prove that the property P is invariant, we must show that the property is true in all initial states and that it is preserved by any state transition.

For the base case, the property P is trivially true since $b = a$ and $s \cup r = \emptyset$. Now, we show that the statement is preserved by any transition $(a, b, s, r) \rightarrow (a', b', s', r')$ of the state machine. Note that the predicate is vacuously true if the initial list a does not contain a majority key. Letting x denote the majority key of a, we analyze the various cases:

Case 1: x is the majority element in the list b.

We we analyze the post-state of the transition by considering the cases in which b has one

or more elements. If b has one element and that element is x, then r' will be a singleton list containing x, *i.e.*, $x \in s' \cup r'$. If b has more than one elements, it suffices to show that if b has an even number of elements, x is the majority element of b', and otherwise, either x is the last element of b and is an element of s', or x is not the last element of b and is the majority element of b'.

Let b contain an even number of $2k$ records and i be the count of x in b, *i.e.*, $i \geq k+1$. Let l and d denote the number of pairs that have same and different elements respectively; that is, $2k = 2l + 2d$. Moreover, let l' and d' denote the number of pairs of same and different elements, respectively, that contain x; that is, $i = 2l' + d'$. In the post-state of the transition, the number of records is $2k - l - 2d$ and the count of the majority element x is $i - l' - d'$. Substituting in for k and i, we get that the number of records and the count of x in b' is l and l', respectively. In order to show that x is the majority element of b', it suffices to show that $2l' \geq l+1$. Consider the inequality $i \geq k+1$ and plug in $i = 2l' + d'$ and $2k = 2l + 2d$. It follows that $2l' + d' \geq l + d + 1$. But since $d \geq d'$, it follows that $2l' + d' \geq l + d + 1 \geq l + d' + 1$. Thus, it is the case that $2l' \geq l + 1$. Thus, the element x is the majority element of b'.

Let b be a list of an odd number of $2k+1$ records. If the last key is not x, then x occurs at least $k+1$ times in the first $2k$ records, *i.e.*, x is a majority key of the first $2k$ elements in b and the prior analysis applies; that is, x is the majority element of b'. If x is the last element of b, then it is appended to the list s; that is, x is an element of the list s'.

Case 2: x is an element of s.
The list s is only updated in two ways: i) an element of s is determined not to be the majority element, in which case that element is discarded, and ii) the majority element is discovered, r' is set to the singleton list containing the majority element, and s' is set to $\emptyset$, *i.e.*, $x \in s' \cup r'$. In the former case, since x is the majority key, it does not get discarded ans, thus, is an element of s'. In the latter case, x is the majority element, so $x \in r'$. It follows that, $x \in s' \cup r'$.

Case 3: x is an element of r.
In this case, no transition is enabled.

Thus, by the above case analysis, we have shown that in the post-state of the transition, x is either the majority key of b' or is in the set $s' \cup r'$, as needed.

In the above analysis, we have proven that the property P is indeed invariant. Moreover, it is simple to see that the state machine terminates either when the sets b and s are both empty, or the set r is non-empty; that is, when no more state transitions are enabled. At that point, since $b = s = \emptyset$ the invariant implies that the majority key must be in r if it indeed existed in the initial list of records a.

□

(b) Write down a recurrence for worst-case time this algorithm takes. For this part assume that the size of the working set at every stage is even.

Solution.

A recurrence for the worst-case time for this algorithm is $T(n) = T(n/2) + n/2$. It is important to also specify, the boundary valuations of the running time; that is, the values for $T(0)$ and $T(1)$. According to the algorithm, if we end up with 0 records, we do nothing and if we end up with one record, we check if the key of that record is the majority key of the initial list of records; that is, $T(0) = 0$ and $T(1) = n$.

□

(c) Solve the recurrence you wrote down in part (b) to obtain an asymptotic time bound (Θ) for worst-case running time. (We are interested in the *order* of growth, not the exact constant.)

Solution.

The asymptotic time bound for the worst-case running time is $T(n) = T(n/2) + n/2 = T(n/4) + n/4 + n/2 = ... = 2n - 3 = \Theta(n)$.

□

(d) Argue that this time bound also holds without the assumption that the working set at each stage is even.

Solution.

If the set is odd, then $T(n) = T(n/2) + n/2 + n = T(n/2) + 3/2n$ which comes down to $T(n) = 4n - 9 = \Theta(n)$.

□

Problem 7 Linear Recurrences

Find closed-form solutions to the following linear recurrences.

(a) $x_n = 5x_{n-1} - 6x_{n-2} \quad (x_0 = 0, x_1 = 1)$

Solution.

The characteristic equation is $r^2 - 5r + 6 = 0$. The roots can be found with the quadratic formula:

$$\begin{aligned} r_1 &= 2 \\ r_2 &= 3 \end{aligned}$$

This implies a solution of the form

$$x_n = A2^n + B3^n.$$

Substituting the initial conditions into this general form gives a system of linear equations:

$$\begin{aligned} 0 &= A + B \\ 1 &= 2A + 3B \end{aligned}$$

The solution to this system is $A = -1$ and $B = 1$. Therefore, the final solution to the recurrence is

$$x_n = 3^n - 2^n.$$

□

(b) $x_n = 5x_{n-1} - 8x_{n-2} + 4x_{n-3} \quad (x_0 = 1, x_1 = 2, x_2 = 3)$

Solution.

The characteristic equation is $r^3 - 5r^2 + 8r - 4 = 0$. Solving a cubic equation can be messy process, but in this case the roots are easy to find:

$$\begin{aligned} r_1 &= 2 \\ r_2 &= 2 \\ r_3 &= 1 \end{aligned}$$

The general form of the solution is therefore

$$x_n = A2^n + Bn2^n + C.$$

Substituting the initial conditions into this general form gives a system of linear equations.

$$\begin{aligned} 1 &= A + C \\ 2 &= 2A + 2B + C \\ 3 &= 4A + 8B + C \end{aligned}$$

The solution to this linear system is $A = 2$, $B = -\frac{1}{2}$, and $C = -1$. The complete solution to the recurrence is therefore

$$x_n = 2^{n+1} - n2^{n-1} - 1.$$

(c) (Fibonacci Plus) $x_n = x_{n-1} + x_{n-2} + 1 \quad (x_0 = 0, x_1 = 1)$

Solution.

First, we find the general solution to the homogeneous recurrence. The characteristic equation is $r^2 - r - 1 = 0$. The roots of this equation are:

$$\begin{aligned} r_1 &= \frac{1+\sqrt{5}}{2} \\ r_2 &= \frac{1-\sqrt{5}}{2} \end{aligned}$$

Therefore, the solution to the homogeneous recurrence is of the form

$$x_n = A\left(\frac{1+\sqrt{5}}{2}\right)^n + B\left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Next, we need a particular solution to the inhomogeneous recurrence. Since the inhomogeneous term is constant, we guess a constant solution b.

$$\begin{aligned} b &= b + b + 1 \\ b &= -1 \end{aligned}$$

Therefore, the inhomogeneous recurrence has the simple particular solution $x_n = -1$. The complete solution to the recurrence is the homogeneous solution plus the particular solution:

$$x_n = A\left(\frac{1+\sqrt{5}}{2}\right)^n + B\left(\frac{1-\sqrt{5}}{2}\right)^n - 1$$

All that remains is to find the constants A and B. Substituting the initial conditions gives a system of linear equations.

$$\begin{aligned} 0 &= A + B - 1 \\ 1 &= A\left(\frac{1+\sqrt{5}}{2}\right) + B\left(\frac{1-\sqrt{5}}{2}\right) - 1 \end{aligned}$$

The solution to this linear system is:

$$\begin{aligned} A &= \frac{5+3\sqrt{5}}{10} \\ B &= \frac{5-3\sqrt{5}}{10} \end{aligned}$$

Therefore, the complete solution to the recurrence is

$$x_n = \left(\frac{5+3\sqrt{5}}{10}\right) \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{5-3\sqrt{5}}{10}\right) \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n - 1.$$

□

Problem 8 Injections, Surjections, and Bijections

Let f be a function from A to B, and g a function from B to C. The composition $g \circ f$ is defined by: $g \circ f(x) = g(f(x))$.

Determine whether each of the following statements is true or false. If it's true, prove it. If it's false, give a counterexample.

(a) If f and g are both injective then $g \circ f$ is injective.

Solution.

Let $a, a' \in A$, $b = f(a)$, $b' = f(a')$, $c = g \circ f(a)$, and $g \circ f(a')$. It suffices to show that if $a = a'$ then $c = c'$. Since f is injective, if $a = a'$ then $b = b'$. Similarly since g is injective, if $b = b'$ then $c = c'$. It follows that, if $a = a'$ then $c = c'$.

□

(b) If f and g are both surjective then $g \circ f$ is surjective.

Solution.

Since g is surjective, it is the case that for all $c \in C$ there exists an element $b \in B$ such that $g(b) = c$. Similarly, since f is surjective, it is the case that for all $b \in B$ there exists an element $a \in A$ such that $f(a) = b$. Thus, it follows that for all $c \in C$, there exists some $a \in A$ such that $g \circ f(a) = c$; that is, $g \circ f$ is surjective.

□

(c) If f is surjective and $g \circ f$ is injective then g is injective.

Solution.

The proof is by contradiction. Let f be surjective and $g \circ f$ be injective and suppose that g is *not* injective. Then, since g is not injective, there exist $b, b' \in B, b \neq b'$ such that $g(b) = g(b')$. Moreover, since f is surjective there exist $a, a' \in A, a \neq a'$ such that $f(a) = b$ and $f(a') = b'$. It follows that, $g \circ f(a) = g \circ f(a')$ which implies that $g \circ f$ is not injective; that is, we have reached a contradiction. It follows that our initial assumption must be false; that is, it is the case that g is injective.

□

(d) If g is injective and $g \circ f$ is surjective then f is surjective.

Solution.

The proof is by contradiction. Let g be injective and $g \circ f$ be surjective and suppose that f is *not* surjective. Since g is injective, it is the case that $|B| \leq |C|$. Since f is not surjective, there exists $b \in B$ for which there does not exist $a \in A$ such that $f(a) = b$; that is, the image of f is a set $B' \subset B$, *i.e.*, $|B'| < |B| \leq |C|$. It follows that $g \circ f$ can not be surjective; that is we have reached a contradiction. It follows that our initial assumption must be false; that is, it is the case that f is surjective.

□

Problem 9 Counting Using Functions

Determine the cardinality of the following:

(a) The number of binary relations on a set A of n elements.

Solution.

Consider the binary matrix representation of a binary relation on a set A of n elements where each element of the matrix indicates whether the corresponding elements of the set A are related. Since each of these matrix elements can be either 0 or 1, the number of binary relations on a set A of n elements is equal to $2^{(n^2)}$.

□

(b) The number of different digraphs without self-loops for a set V of n vertices.

Solution.

Consider the adjacency matrix of a digraph without self-loops for a set V of n vertices. Since the digraph is restricted to have no self-loops, elements on the diagonal of the adjacency matrix are restricted to equal 0. However, the remaining elements of the adjacency matrix can be either 0 or 1. It follows that the number of digraphs without self-loops for a set V of n vertices is equal to $2^{(n^2-n)}$.

□

(c) The number of functions from A to B, where $|A| = n$ and $|B| = m$.

Solution.

Since the image of any element of A is unique, it follows that each element of A could map to any one element of B. Thus, the number of functions from A to B, where $|A| = n$ and $|B| = m$, is equal to m^n.

□

Problem 10 Pigeonhole Principle

(a) Prove that among every set of 18 integers, there exists a pair with sum or difference divisible by 32.

Solution.

We assign each of the 18 numbers to one of 17 bins, defined as follows:

First, we define 15 "ordinary" bins: bin 1 contains numbers congruent to 1 or 31 modulo 32, bin 2 contains numbers congruent to 2 to 30 modulo 32, ..., bin 15 contains numbers congruent to 15 or 17 modulo 32. Now, we define two "special" bins: bin 0 contains numbers congruent to 0 modulo 32, and bin 16 contain numbers congruent to 16 modulo 32.

By the Pigeonhole Principle, at least two out of the 18 numbers are assigned to the same bin. If two numbers are assigned to an ordinary bin and are congruent modulo 32, then their difference is divisible by 32. If two numbers are assigned to an ordinary bin and are not congruent modulo 32, then their sum is divisible by 32. If two numbers are assigned to a special bin, then both their sum and their difference is divisible by 32.

□

(b) One of the 6.042 TAs has decided to conduct his next tutorial in the following manner: he will arrange his m students in alphabetical order and hand out tutorial problems to each of the students such that each student ends up with one or more problems. The TA claims that using such an arrangement, there is always a group of (one or more) students that are consecutive in the alphabetical order and collectively receive a number of problems that is a multiple of m.

As an example, consider a 6.042 tutorial with $m = 10$ students in which the students are arranged in alphabetical order and each of which receives $1, 2, 4, 8, 16, 16, 8, 4, 2, 1$ problems. Then, since the fourth, fifth, and sixth students have collectively received 40 problems, the TA's claim holds.

Prove this claim for all m.

Solution.

Let P_i be the total number of problems given out to the first i students. There are two cases. In the first case, suppose that at least one of $P_1, P_2, \ldots, P_m$ is congruent to 0 modulo m. In particular,

suppose that P_j is congruent to 0 modulo m. This implies that the number of problems received by the first j students is a multiple of m, and so the claim holds. In the second case, suppose that none of $P_1, P_2, \ldots, P_m$ is congruent to 0 modulo m. Then each of the m terms $P_1, P_2, \ldots, P_m$ is congruent to one of the $m-1$ values $1, 2, \ldots, m-1$ modulo m. By the Pigeonhole Principle, there are at least two terms P_j and P_k that are congruent to the same value modulo m. Thus students $j+1, j+2, \ldots, k$ received a total of $P_k - P_j$ problems, which is a multiple of m.

$\square$