Massachusetts Institute of Technology
6.042J/18.062J: Mathematics for Computer Science
Professor David R. Karger

Handout 14
March 6, 1997

Problem Set 4 Solutions

This is a rough draft only.

Problem 1. *Let n be a positive integer with no factors (other than 1) that are perfect squares. Let a be the sum of the factors of n. Let b be the sum of the squares of the factors of n. Let c be the sum of the factors of n^3. Prove using well-ordering that $a \cdot b = c$.*

First prove that this is true for n=1: The factors of $n = 1$ are just 1, so $a = 1$, $b = 1$, and $c = 1$, and we see that $a \cdot b = 1$.

Prove by well-ordering that for $n \geq 2$, $a \cdot b = c$:

Let $S = \{n \mid a \cdot b \neq c$ and where n has no factors that are squares (except 1) $\}$. Assuming S is not empty, then by well-ordering, we must have a least element $x \in S$.

By the Fundamental Theorem of Arithmetic, x can be factored into a product of primes. Let p be a prime factor of x (which exists since $x \geq 2$), and set $m = x/p$. Of couse, $m < x$.

Let a', b', and c' be the sum of factors, their squares and the sum of factors of m^3, respectively.

Since a' is the sum of all factors of $m = x/p$ and p is not a prime factor of m (since x has no factors which are squares, each prime factor appears only once in x), $a = a' + p \cdot a'$. (All the factors of m and all new factors that can be made by multiplying the old factors by p.)

Similarly, $b = b' + p \cdot b'$.

Now, $x^3 = m^3 p^3$, so to get all the factors of x^3, we need all factors of m^3 (which is c) and all new combinations that can be made with p^3. This gives us

$$c = c' + pc' + p^2 c' + p^3 c' = c'(1 + p + p^2 + p^3)$$

Substituting for a, b, c in $a \cdot b \neq c$, we get

$$a'(1 + p) \cdot b'(1 + p^2) \neq c'(1 + p + p^2 + p^3)$$

which gives us $a' \cdot b' \neq c'$.

So m must be in the set S as well, but x is the least element of S, and $m < x$ so we have a contradiction, and S must be empty.

Problem 2. *Fill in the missing proof from class, that if $p \mid a_1 a_2 \cdots a_k$, then $(\exists i)(p \mid a_i)$.*

We will prove the theorem by induction.

Recall that we have the following theorem:
If p is prime and $p \mid a \cdot b$ then $p \mid a$ or $p \mid b$.

Hypothesis: $H(i) =$ if p is prime, and $p \mid a_1 \ldots a_k$, then $(\exists j, 1 \leq j \leq i)(p \mid a_j) \vee p \mid a_{i+1} \ldots a_k$.

Base case:
$p \mid a_1 a_2 \cdots a_k$ (given to us as true)

Inductive step: Assume $H(i)$ and show $H(i+1)$.
Applying the thereom to $p \mid a_{i+1} \ldots a_k$ we get

$$(\exists j, 1 \leq j \leq i)(p \mid a_j) \vee p \mid a_{i+1} \ldots a_k$$

$$(\exists j, 1 \leq j \leq i)(p \mid a_j) \vee p \mid a_{i+1} \vee p \mid a_{i+2} \ldots a_k$$
$$(\exists j, 1 \leq j \leq i+1)(p \mid a_j) \vee p \mid a_{i+2} \ldots a_k$$

which satifies H(i+1). So the theorem is proved.

Problem 3. *Show that if a, b, and m are positive integers such that $m \geq 2$ and $m \mid (a-b)$, then $gcd(a, m) = gcd(b, m)$.*

If $m \mid (a-b)$, then there is some q such that $(a-b) = mq$ which can be rewritten as $a = mq + b$. By Euler's method, we know from this that $\gcd(a, m) = \gcd(m, b)$. Since gcd is reflexive, we know that $\gcd(a, m) = \gcd(b, m)$.

Problem 4. *Let a and b be positive integers. The least common multiple of a and b is defined to be the smallest positive integer they both divide. Prove that this quantity always exists. Prove that $a \cdot b = gcd(a, b) \cdot lcm(a, b)$.*

For each distinct prime in the prime factorizations, ab uses the sum of the number of occurrences in a and b, while $\gcd ab \times \mathrm{lcm}(a, b)$ uses the minimum plus the maximum number of occurrences. So for each distinct prime, the number of occurrences is the same (the sum of two numbers is the same as their min plus their max).

Problem 5. *We are given two buckets, A and B, a water hose, a receptacle, and a drain. The buckets and receptacle are initially empty. The buckets are labelled with their capacities, which are the positive integers a and b, respectively. The receptacle can be used to store an unlimited amount of water, but has no measurement markings. Excess water can be dumped into the drain. Consider the following "Die Hard" operations:*

- *fill bucket A or bucket B from the hose*
- *pour from the receptacle to either A or B until it is full or the receptacle is empty (whichever happens first)*
- *empty bucket A or bucket B to the drain or the receptacle*
- *pour from A to B until either A is empty or B is full (whichever happens first)*

- *pour from B to A until either B is empty or A is full (whichever happens first)*

Let $c = gcd(a, b)$. Let g be a positive integer. Prove that we can put g gallons into the receptacle using the above operations if, and only if, $c \mid g$.

Note that the question asks us to prove two implications: one corresponding to "if" and the other corresponding to "only if."

Lemma 0.1 *At the end of any sequence of n steps, every container contains a multiple of c gallons.*

Proof. The base case is for $n = 0$. In this case, the problem statement gives that each container has 0 gallons, which is a multiple of c. Let n be an arbitrary natural number and consider an arbitrary sequence of $n+1$ steps. At the end of the first n steps, (by the inductive hypothesis) every container contains a multiple of c gallons. Since each of the two buckets holds a multiple of c gallons, the amount of room left in each of the buckets is also a multiple of c gallons. Each "Die Hard" operation transfers an amount of water corresponding either to the amount of water in a container just before the operation is performed or the amount of room left in a bucket just before the operation is performed. Thus the amount of water transferred in the $(n + 1)$st operation is a multiple of c gallons. Since neither removing a multiple of c gallons from a container nor adding a multiple of c gallons to a container will change the fact that it contains a multiple of c gallons, every container will still contain a multiple of c gallons after the $(n + 1)$st operation is performed. ∎

By the lemma, if g gallons can be put into the receptacle using the above operations, $c \mid g$.

For the reverse direction, we will also use a lemma.

Lemma 0.2 *If $c \mid g$, there are* positive *integers m and n such that $g = ma - nb$.*

Proof. Since $c \mid g$, there is a positive integer t such that $g = tc$. As shown in class, there are integers r and s such that $c = ra + sb$. Let $m \equiv rt + (|r| + |s| + 1)bt$ and $n \equiv -st + (|r| + |s| + 1)at$. Then

$$\begin{aligned} ma - nb &= (rt + (|r| + |s| + 1)bt)a - (-st + (|r| + |s| + 1)at)b \\ &= tra + tsb + (|r| + |s| + 1)abt - (|r| + |s| + 1)abt \\ &= t(ra + sb) = tc = g. \end{aligned}$$

Since $m = (r + |r|\, b)t + (|s| + 1)bt$, m is clearly positive. Since $n = (-s + |s|\, a)t + (|r| + 1)at$, n is clearly positive. ∎

To complete the reverse direction, suppose $c \mid g$ and let m and n be as in the above lemma. Fill bucket A m times and each time empty the water into the receptacle. At this point the receptacle contains ma gallons of water, of course. Then fill bucket B n times from the receptacle and each time dump the water into the drain. Note that there is always enough water to do this since $ma - nb = g > 0$ and the receptacle has some multiple of b gallons more than this before each filling of bucket B. Once this is done, the receptacle contains exactly $ma - nb = g$ gallons of water.

Therefore, whenever $c \mid g$ we can put g gallons into the receptacle using the above operations.

Problem 6. *Use Euclid's algorithm to compute $c = gcd(171, 63)$ and compute integers x and y for which $171x + 63y = c$ (show your work). Using the model of the "Die Hard" problem, give an explicit scheme for putting exactly 18 ounces of water into the receptacle (the final holder) using a 171 ounce and a 63 ounce bucket.*

Performing Euclid's algorithm:

$$171 = 2 * 63 + 45$$
$$63 = 1 * 45 + 18$$
$$45 = 2 * 18 + 9$$
$$18 = 2 * 9 + 0$$

Thus gcd(171, 63) = 9. We can rewrite the first three equations as follows:

$$45 = 171 + -2 * (63)$$
$$18 = 63 + -1 * (45)$$
$$9 = 45 + -2 * (18)$$

Substituting the second into the third gives:

$$9 = 45 + -2 * (63 + -1 * (45)) = 45 + -2 * (63) + 2 * (45) = -2 * (63) + 3 * (45)$$

Substituting the first into this gives:

$$9 = -2 * (63) + 3 * (171 + -2 * (63)) = -2 * (63) + 3 * (171) + -6 * (63) = 3 * (171) + -8 * (63)$$

So $18 = 2 * 9 = 2 * (3 * (171) + -8 * (63)) = 6 * (171) + -16 * (63)$. Thus x can be 6 and y can be -16.

This means that one way of putting exactly 18 ounces of water into the receptacle is to first put 1026 ounces of water in the receptacle by filling the 171 ounce bucket 6 times and emptying it into the receptacle each time and then remove the extra 1008 ounces by filling the 63 ounce bucket 16 times and emptying it into the drain each time.

Note that this is not the most efficient way to put exactly 18 ounces of water into the receptacle, since $18 = 3 * (63) + -1 * (171)$. Thus we can fill the 63 ounce bucket three times and empty it into the receptacle each time and then fill the 171 ounce bucket from the receptacle and empty it into the drain.

Problem 7. *Prove the correctness of the following algorithm for calculating m^n. Assume that initially, for any call to Power(m, n), $n \geq 1$.*

procedure Power(m, n)

if $(n = 1)$

    return(m)

else

    return(Power$(m, \lfloor n/2 \rfloor)*$ Power $(m, n - \lfloor n/2 \rfloor))$

(note that $\lfloor x \rfloor$ truncates to an integer, so $\lfloor 5/2 \rfloor = 2$. See Rosen p. 68)

Proof: To prove the correctness of a procedure, we need to show tht the procudure returns the correct value, and also that it terminates.

Show the code returns the corect value, proof by induction:

$H(n) =$ Power(m,j) returns m^j.

Base case:
Power (m,1) returns $m = m^1$.

Inductive step:
Assuming $H(i)$ is true for all $1 \leq i < n$. Power(m,n) returns

$$\text{Power}(m, \lfloor n/2 \rfloor) * \text{Power}(m, n - \lfloor n/2 \rfloor)$$

Since $\lfloor n/2 \rfloor < n$ and $n - \lfloor n/2 \rfloor < n$, by the inductive hypothesis, Power(m,n) returns $m^{\lfloor n/2 \rfloor} \cdot (m^{n - \lfloor n/2 \rfloor} = m^n$. So the hypothesis is true.

Now, show termination:
Let $H(n) = P(m, n)$ terminates

Base case:
$P(m, 1)$ terminates when it returns.

Inductive step:
Assume $P(m, n)$ terminates for $i < n$. Power (m,n) calls Power$(m, \lfloor n/2 \rfloor)$ and Power$(m, n - \lfloor n/2 \rfloor))$ which by the inductive hypothesis, both terminate. So, Power(m,n) terminates after an additional multiplication,

Problem 8. *Prove that the following algorithm for computing the Fibonacci numbers is correct.*

procedure Fib(n)

if $n = 0$ then $y := 0$

else

    $x := 0$

$y := 1$

for $i := 1$ to $n-1$

$z := x + y$

$x := y$

$y := z$

end for

return(y)

(y is the nth Fibonacci number).

Proof: Show that Fib(n) returns the nth fibonacci number.

Hypothesis: Fib(n) returns the nth fibonacci number.

Proof by cases:
Fib(0) returns 0=fib(0) by the first if statement
Fib(1) returns 1=fib(1) by the second if statement, while not enterring the for loop
Fib(n) where $n \geq 2$:
We will inductively show that the loop calculates the appropriate number.

H(i): After the ith iteration of the for loop, $x = fib(i)$ and $y = fib(i+1)$.

Base case:
H(1): $x = 0$ and $y = 1$ upon entering the loop. Upon exitting, $z = 1$, $x = 1 = fib(1)$, and $y = 1 = fib(2)$.

Inductive step:
Assuming H(i) is true for all $i < m$, on entering the mth iteration of the loop, $x = fib(m)$, and $y = fib(m+1)$ by the inductive hypothesis. After finishing the loop, $x = y_{prev} = fib(m)$, and $y = z = fib(m) + fib(m+1) = fib(m+2)$. So H(i) is true for all $i \geq 1$.

Since the for loop goes through $n-1$ iterations, the loop finishes with $y = fib(n)$, and y is then returned. So for all $n \geq 2$, Fib(n) returns the right value.

Now, show that Fib(n) terminates:
This is pretty simple. If statements always terminate, and the for loop will terminate because neither i nor n are changed in the loop.

Problem 9. **(Optional).** *Consider the "Die Hard" problems. Can you solve problems 5 and 6 if you are forbidden to ever remove water from the receptacle?*

Yes. For problem 5, the proof that you can only put g gallons of water into the receptacle if $c \mid g$ covers all methods, whether they remove water from the receptacle or not. Thus you need only prove that you can put g gallons of water into the receptacle without ever removing water from the receptacle if $c \mid g$. To prove this, we will again make use of a lemma. Without loss of generality (WOLOG), we will say $a \geq b$.

Lemma 0.3 *If k is a natural number and j is the smallest natural number such that $ja - kb \geq 0$, then you can put $ja - kb$ gallons of water into bucket A using the "Die Hard" operations without ever using the receptacle.*

Proof. Base Case: $k = 0$. Then $j = 0$. You don't have to do anything to put $ja - kb = 0$ gallons of water into bucket A.

Inductive Step: Let k be an arbitrary natural number. Suppose j' is the smallest natural number such that $j'a - (k+1)b \geq 0$. Then $j'a - kb \geq 0$. So by well-ordering, there is a smallest natural number j for which $ja - kb \geq 0$. By the inductive hypothesis, you can put $ja - kb$ gallons of water into bucket A using the "Die Hard" operations without ever using the receptacle. We consider two cases.

Case 1: $j = j'$. After putting $j'a - kb$ gallons of water into bucket A (as we have just assumed that you can do), you can empty bucket B into the drain and then fill bucket B from bucket A, leaving $j'a - (k+1)b$ gallons of water in bucket A.

Case 2: $j < j'$. It is easy to see that $ja - kb < b$. Then after putting $ja - kb$ gallons of water into bucket A, you can empty out bucket B and put all $ja - kb$ gallons of water into bucket B. Then you can fill bucket A from the hose and pour from bucket A into bucket B until bucket B is full (since $a \geq b$). At this point the total amount of water in buckets A and B is $(j+1)a - kb$ gallons. Since bucket B is full, bucket A contains exactly $(j+1)a - (k+1)b$ gallons of water. Since this quantity is nonzero, it is clear that $j' = j + 1$. So bucket A contains $j'a - (k+1)b$ gallons of water.

■

Suppose $c \mid g$. By our second lemma from the solution to problem 5, we know that there are positive integers m and n such that $g = ma - nb$. Since m is a natural number such that $ma - nb \geq 0$, there is a smallest natural number j for which $ja - nb \geq 0$ (by well-ordering). By the above lemma, you can put $ja - nb$ gallons of water into bucket A without using the receptacle. Then you can empty bucket A into the receptacle and fill bucket A from the hose $m - j$ times, emptying it into the receptacle each time. (Why is $m - j > 0$?) This gives $g = ma - nb$ gallons in the receptacle.

It is an easy exercise to prove that for any natural number k there really is a smallest natural number j such that $ja - kb \geq 0$, but we don't really have to prove this in order for our revised solution to problem 5 to work. Why not?

The revised solution to problem 5 provides a revised solution to problem 6 as well.