Fall '15 CS 5800 Algorithms	Hackerrank Group: singhay

Problem Set 1

Surekha Jadhwani, Akash Singh, and Ayush K. Singh

1 Part 1

Answers

1. **Given:** A program P which computes the function $f(n) = 1 for$ for all $n \in N$ and an arbitrary program Q.
   To Prove: There exists another program R which determines whether P(n) = Q(n) for any input n.
   Proof: Proof by contradiction.
   Assume that a Program R exists which could determine equivalence of two programs.
   Now, consider a Program P which returns always 1.
   i.e. Program P is something like:
   Program P
   return 1;
   end Now consider a Program R which takes input program Q and returns whether or not the programs are equal.
   Program R(Program Q)
   return P == Q;
   end
   Say we pass program Q which works on an integer input into variable i and is as follows:
   Program Q(int i)
   if (i == 1)
   return 1;
   else
   loop forever;
   end
   Hence if a program Q in above format is passed, the program in many cases will never return an output.
   In fact, any program which does not halt, if passed as an input, equivalence cannot be deduced.
   For a program to be able to compute whether two programs are equal or not, the programs passed should always halt.
   But, since we can never say whether a program will halt or not as per the proof of halting problem, we know that a program which can be used to check equivalence cannot exist.

2. **Modular Arithmetic**

(a) In a modulo 13 system, what is the multiplicative inverse of 15?

Solution: Let x be the multiplicative inverse of 15 in a modulo 13 system.

By Extended Euclidean Method

$13 = 15 \cdot 0 + 13 \cdot 1$

$15 = 13 \cdot 1 + 2 \cdot 1$

$13 = 2 \cdot 6 + 1$

$13 - 2 \cdot 6 = 1$

$13 - (15 - 13) \cdot 6 = 13 + 13 \cdot 6 - 15 = 1$

$13 \cdot \mathbf{7} - 15 = 1$

Answer: $x = 7$

(b) To what number between 0 and 12 inclusive is the product 3·5·11·17·23·29·31·47·53 congruent modulo 13?
Solution:$x \equiv (3 \cdot 5 \cdot 11 \cdot 17 \cdot 23 \cdot 29 \cdot 31 \cdot 47 \cdot 53) mod 13$
$x mod 13 = (3 \cdot 5 \cdot 11 \cdot 17 \cdot 23 \cdot 29 \cdot 31 \cdot 47 \cdot 53) mod 13$
$x mod 13 = ((3 mod 13)(5 mod 13)(17 mod 13)(11 mod 13)(23 mod 13)(29 mod 13)..$
$..(31 mod 13)(47 mod 13)(53 mod 13) mod 13)$
$x mod 13 = (3 \cdot 5 \cdot 4 \cdot 11 \cdot 10 \cdot 3 \cdot 5 \cdot 8 \cdot 1) mod 13$
$x mod 13 = (15 \cdot 44 \cdot 30 \cdot 40) mod 13$
$x mod 13 = (2 \cdot 5 \cdot 4 \cdot 1) mod 13$
$x mod 13 = 40 mod 13$
Answer: $x = 1$

(c) Find the remainder when 19^{19} is divided by 11
Precompute: $19^1 mod 11 = 8$ (Equation 1)
$19^2 mod 11 = (19^1 mod 11 \cdot 19^1 mod 11) mod 11$
$= (8 \cdot 8) mod 11 = 9$ [(from Eq. 1)(Equation 2)]
$19^4 mod 11 = [19^2 mod 11 \cdot 19^2 mod 11] mod 11$
$= (9 \cdot 9) mod 11 = 4$ [(from Eq. 2)(Equation 3)]
$19^8 mod 11 = [19^4 mod 11 \cdot 19^4 mod 11] mod 11$
$= (4 \cdot 4) mod 11 = 5$ [(from Eq. 3)(Equation 4)]
$19^{16} mod 11 = [19^8 mod 11 \cdot 19^8 mod 11] mod 11$
$= (5 \cdot 5) mod 11 = 3$ [(from Eq. 4)(Equation 5)]
Solution: We know that, $19^{19} = 19^{16} \cdot 19^2 \cdot 19^1$ Therefore from Eq. 1, Eq. 2, and Eq. 3, We can write
$19^{19} mod 11 = (19^{16} mod 11 \cdot 19^2 mod 11 \cdot 19^1 mod 11) mod 11$
$= (8 \cdot 9 \cdot 3) mod 11 = (8 \cdot (9 \cdot 3)) mod 11$
$= (8 mod 11 \cdot 27 mod 11) mod 11$
$= (8 \cdot 5) mod 11$
$= 40 mod 11$
Answer: $19^{19} mod 11 = 7$

3. Solve the following using Mathematical Induction

(a) For any integer $n \geq 0, 3 | (2_{2n} - 1)$
To Prove: $3 | (2_{2n} - 1) i.e (2^{2n} - 1)/3 = k$, an integer OR $(2^{2n} - 1) = 3k$
Proof: by Mathematical Induction
Let $P(n)$ be $(2^{2n} - 1) = 3k$
Basic Step: for $n = 1$
$P(1) : (2^{2*1} - 1) = 3$
Induction Step: Since $P(n)$ is true for $n = 1$, assuming its true for $P(n+1)$
$P(n+1) : (2^{2(n+1)} - 1) = (2^{2n}.2^2 - 1)$
$P(n+1) : [(3k+1)2^2 - 1] = 12k + 4 - 1$
$P(n+1) : 12k + 3 = 3(4k+1)$ $P(n+1) := 3j$ [where j = 4k + 1]
Hence Proved.

(b) $1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \cdots + \frac{1}{\sqrt{n}} \leq 2\sqrt{n}$, for all $n \geq 1$

To Prove: $1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \cdots + \frac{1}{\sqrt{n}} \leq 2\sqrt{n}$, for all $n \geq 1$

Proof: By Mathematical Induction

Let $P(n)$ be $1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \cdots + \frac{1}{\sqrt{n}} \leq 2\sqrt{n}$

Basic Step: For $n = 1$

$P(1)$: $1 + \frac{1}{1} \leq 2\sqrt{1}$

$P(1)$: $1 \leq 1$

Induction Step: Since $P(n)$ is true for $n = 1$, assuming its true for $P(n)$

Solving for $P(n+1)$

$P(n+1)$: $1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \cdots + \frac{1}{\sqrt{n}} + \frac{1}{\sqrt{n+1}} \leq 2\sqrt{n+1}$

Replacing value for $P(n)$ in above expression

$P(n+1)$: $2\sqrt{n} + \frac{1}{\sqrt{n+1}} \leq 2\sqrt{n+1}$

Multiplying both sides by $\sqrt{n+1}$

$P(n+1)$: $2\sqrt{n}\sqrt{n+1} + 1 \leq 2(n+1)$

$P(n+1)$: $2\sqrt{n^2+n} + 1 \leq 2n + 2$

Subtracting 1 from both sides

$P(n+1)$: $2\sqrt{n^2+n} \leq 2n + 1$

Squaring both sides

$P(n+1)$: $4(n^2 + n) \leq (2n+1)^2$

$P(n+1)$: $4n^2 + 4n \leq 4n^2 + 4n + 1$

Canceling out common terms

$P(n+1)$: $0 \leq 1$

Hence Proved.

(c) Let n and k be non-negative integers with $n \geq k$.

To Prove: $\sum_{i=k}^{n} \binom{i}{k} = \binom{n+1}{k+1}$, for $n, k \geq 0$ and $n \geq k$

Proof: By Mathematical Induction

Let $P(n)$ be $\sum_{i=k}^{n} \binom{i}{k} = \binom{n+1}{k+1}$

We know that $\binom{n}{k} = \frac{n!}{(n-k)!k!}$

Basic Step: For k=1,n=2

$P(1)$: $\sum_{i=1}^{2} \binom{i}{1} = \binom{2+1}{1+1}$

$P(1)$: $\binom{1}{1} + \binom{2}{1} = \binom{3}{2}$
$P(1)$: $\frac{1!}{0!1!} + \frac{2!}{1!1!} = \frac{3!}{2!1!}$
$P(1)$: $\frac{1}{1} + \frac{2}{1} = \frac{6}{2}$
$P(1)$: $3 = 3$
Inductive Step: Since $P(n)$ is true for $n = 1$, assuming its true for $P(n)$
Solving for $P(n + 1)$
$P(n + 1)$: $\sum_{i=k}^{n+1} \binom{i}{k} = \binom{n+1+1}{k+1}$
$P(n + 1)$: $\sum_{i=k}^{n} \binom{i}{k} + \binom{n+1}{k} = \binom{n+2}{k+1}$
Replacing value for $P(n)$ in above expression
$P(n + 1)$: $\binom{n+1}{k+1} + \binom{n+1}{k} = \binom{n+2}{k+1}$
$P(n + 1)$: $\frac{(n+1)!}{(n+1-k-1)!(k+1)!} + \frac{(n+1)!}{(n+1-k)!k!} = \frac{(n+2)!}{(n+2-k-1)!(k+1)!}$
$P(n + 1)$: $\frac{(n+1)!}{(n-k)!(k+1)!} + \frac{(n+1)!}{(n+1-k)!k!} = \frac{(n+2)!}{(n+1-k)!(k+1)!}$

Simplifying by expanding factorial and equalizing denominators, we get
$P(n + 1)$: $\frac{(n-k+1)(n+1)!}{(n-k+1)(n-k)!(k+1)k!} + \frac{(k+1)(n+1)!}{(n-k+1)(n-k)!(k+1)k!}$
$= \frac{(n+2)(n+1)!}{(n-k+1)(n-k)!(k+1)k!}$

Canceling out common terms
$P(n + 1)$: $(n - k + 1) + (k + 1) = (n + 2)$
$P(n + 1)$: $n + 2 = n + 2$
$P(n + 1)$: $0 = 0$
Hence Proved.

4. Functions in ascending order of growth rate
Solution: Applying logarithms and simplifying given functions, we get:

No.	**g(x)**	**log(g(x))**
1	$n^{101/100}$	$1.01 \cdot \log n$
2	$n \cdot 2^{n+1}$	$\log n + (n+1)(\log 2)$
3	$n(log\text{n})^3$	$\log n + 3 \cdot \log(\log n)$
4	$n \log n$	$\log n + \log\log n$
5	$n^{loglogn}$	$\log\log n \cdot \log n$
6	$\log(n^{2n})$	$\log\log n^{2n}$
7	n^{logn}	$\log n \cdot \log n$
8	2^n	$n \cdot \log 2$
9	$n \cdot 2^n$	$\log n + n \cdot \log 2$
10	$2^{\sqrt{logn}}$	$\log 2 \cdot sqrtlogn$
11	$2^{2^{n+1}}$	$2^{n+1} \cdot \log 2$
12	e^{e^n}	$e^n \cdot (\log e)$
13	$\log(n!)$	$\log n + \log\log n$
14	e^{logn}	$\log n \cdot \log e$
15	$2^{log(\sqrt{n})}$	$\frac{1}{2} \cdot \log 2 \cdot \log n$
16	$\sqrt{2^{logn}}$	$\frac{1}{2} \cdot \log 2 \cdot \log n$
17	2^{n^2}	$n^2 \cdot \log 2$
18	$n!$	$n \log n$
19	$(\log n)!$	$\log n \cdot \log\log n$
20	$\log\log n$	$\log\log\log n$

By comparing $\log(g(x))$ values for all given $g(x)$ from the above table, we get can write the functions in ascending order of complexity as follows:
$g_{20} < g_{10} < g_{15} = g_{16} < g_{14} < g_1 < g_{19} < g_5 \leq g_{13} < g_4 \leq g_6 < g_3 < g_7 < g_8 < g_9 < g_2 < g_{18} < g_{17} < g_{11} < g_{12}$

5. For a given two functions f, g and h. Decide whether each of the following statements are correct and give a proof for each part

   (a) If $f(n) = \Omega(g(n))$ and $g(n) = \Omega(f(n))$, then $f(n) = \Theta(g(n))$.
   Solution: The above statement is correct and the proof is as follows:
   Given: $f(n) = \Omega(g(n))$, $g(n) = \Omega(f(n))$
   To Prove: $f(n) = \Theta(g(n))$
   Proof: We have $f(n) = \Omega(g(n))$
   By definition of Big-Omega notation, it implies
   $\exists\ c_1>0,\ n_0 \geq 0\colon \forall\ n>n_0$:
   $f(n) \geq c_1 g(n)$ (Equation 1)

   Also, $g(n) = \Omega(f(n))$
   i.e., $\exists\ c_2>0,\ n_0 \geq 0\colon \forall\ n>n_0$: $g(n) \geq c2f(n)$

i.e., $1/c_2 * (g(n)) \geq f(n)$ [As $c_2 > 0$]
Let $1/c_2 = c$ (some constant)
Therefore, $cg(n) \geq f(n)$
(Equation 2)
From Equations (1) and (2), we get
$\exists\ c_1>0,\ n_0 \geq 0:\ \forall\ n>n_0:$
$f(n) \geq c_1 g(n)$ (Equation 1)

Also, $g(n) = \Omega(f(n))$
i.e., $\exists\ c_2>0,\ n_0 \geq 0:\ \forall\ n>n_0:\ g(n) \geq c2f(n)$ i.e., $1/c_2 * (g(n)) \geq f(n)$ [As $c_2 > 0$]
Let $1/c_2 = c$ (some constant)
Therefore, $cg(n) \geq f(n)$ (Equation 2)

From Equations (1) and (2), we get
$cg(n) \geq f(n) \geq c_1 g(n)$ where c, $c_1>0$ (Equation 3)
We know that, $f(n) = \Theta(g(n))$
iff $\exists\ c_1,c_2>0,\ n_0 \geq 0:\ \forall\ n>n_0:\ c_1 g(n) \geq f(n) \geq c_2 g(n)$ (Equation 4)

From equations (3) and (4), we conclude that given f(n) = $\Theta(g(n))$ if $f(n) = \Omega(g(n))$ and $g(n) = \Omega(f(n))$. Hence, proved.

(b) If $f(n) = (g(n))$, then $g(n) \notin O(f(n))$
Solution: The above statement is correct and the proof is as follows:
Given: $f(n) = o(g(n))$
To Prove: $g(n) \in O(f(n))$
Proof: We have $f(n) = o(g(n))$
By definition of Little-o notation, it implies
$\forall\ c_1 > 0:\ \exists n_0 \geq 0:\ \forall\ n>n_0 f(n)< c_1 g(n)$ (Equation 1)
Lets assume, $g(n) \in O(f(n))$
i.e., $\exists c_2>0,\ n_0 \geq 0:\ \forall\ n>n_0:\ g(n) \leq c_2 f(n)$ (Equation 2)
We can see that Equation (2) contradicts with Equation (1).
Hence, Proving by contradiction $g(n) \notin O(f(n))$ if $f(n) = o(g(n))$

(c) If f(n) = O(h(n)) and g(n) = O(h(n)), then f(n) + g(n) = O(h(n))
Solution: The above statement is correct and the proof is as follows:
Given: $f(n) = O(h(n))$ and $g(n) = O(h(n))$
To Prove: $f(n) + g(n) = O(h(n))$
Proof: We have $f(n) = O(h(n))$
By definition, $\exists\ c_1>0,\ n_0 \geq 0:\ \forall\ n>n_0:\ f(n) \leq c_1 h(n)$ (Equation 1)
Also, $g(n) = O(h(n))$
By definition, $\exists\ c_2>0,\ n_0 \geq 0:\ \forall\ n>n_0:\ f(n) \leq c_2 h(n)$ (Equation 2)
L.H.S. = $f(n) + g(n) \leq c_1 h(n) + c_2 h(n)$ [From 1 and 2]
L.H.S. = $(c_1 + c_2)h(n)$
L.H.S. = $ch(n)$ [where $c = c_1 + c_2$]

Therefore, $f(n) + g(n) \leq \mathrm{c}h(n)$
i.e. $f(n) + g(n) = O(h(n))$ Hence Proved.

(d) If $f(n) = O(h(n))$ and $g(n) = O(h(n))$, then $f(n) \cdot g(n) = O(h(n))$
Solution: The above statement is incorrect and the proof is as follows:
Given: $f(n) = O(h(n))$ and $g(n) = O(h(n))$
To Prove: $f(n) \cdot g(n) = O(h(n))$
Proof: We have $f(n) = O(h(n))$
By definition, $\exists$ c_1>0, n_0 ≥0: $\forall$ n>n_0: $f(n) \leq c_1 h(n)$ (Equation 1)
Also, $g(n) = O(h(n))$
By definition, $\exists$ c_2>0, n_0 ≥0: $\forall$ n>n_0: $f(n) \leq c_2 h(n)$ (Equation 2)
L.H.S. = $f(n) \cdot g(n) \leq c_1 h(n) \cdot c_2 h(n)$ [From 1 and 2]
L.H.S. = $(c_1 \cdot c_2) h(n)^2$
L.H.S. = $\mathrm{c}h(n)^2$ [where c = $c_1 \cdot c_2$]
Therefore, $f(n) \cdot g(n) \leq \mathrm{c}h(n)^2$ [$h(n) \neq h(n)^2$]
but $f(n) \cdot g(n) < O(h(n)^2)$ Hence Statement is False.

6. Justify and give the asymptotic upper and lower bounds for each of the following recurrences

(a) $T(n) = 3T(n/2) + n/logn$
Solution: By Akra-Bazzi Method.

$$T(x) = g(x) + \Sigma_{i-1}^{k} a_i T(b_i x + h_i(x)) \text{ for } x \geq x_0$$

The conditions for usage are:

- sufficient base cases are provided
- a_i and b_i are constants for all i
- a_i >for all i
- 0<b_i<1 for all i
- $|g(x)| \in O(x^c)$, where c is a constant and O notates Big O notation
- $|h_i(x)| \in O\left(\frac{x}{(\log x)^2}\right)$ for all i
- x_0 is a constant

The asymptotic behavior of T(x) is found by determining the value of p for which $\sum_{i=1}^{k} a_i b_i^p = 1$ and plugging that value into the equation.

$$T(x) \in \Theta\left(x^p \left(1 + \int_1^x \frac{g(u)}{u^{p+1}} du\right)\right)$$

Here, $a = 3$, $b = \dfrac{1}{2}$, $g(u) = \dfrac{n}{logn}$, $p =??$

Applying that in our situation, $3 \cdot \left(\dfrac{1}{2}\right)^p = 1 \Rightarrow p = 1.58496$

Substituting in the above mentioned Equation,

$$T(x) \in \Theta\left(x^{1.58496}\left(1 + \int_1^x \frac{u/logu}{u^{1.58496+1}} du\right)\right)$$

On Solving and Integrating the later part of above Equation for n we get,

$$T(n) \in \Theta\left(n(loglogn)\right)$$

Now we actually can't use the master method to solve this recurrence relation because of non polynomial difference between $f(n)$ and $n^{log_2 3}$. However, we can still derive an upper bound by finding a similar recurrence that is larger than T(n), analyze the new recurrence using the master method, and use the result as an upper bound for T(n). $T(n) = 3T(n/2) + n/logn \leq 3T(n/2) + n$, so if we call $S(n)$ the function such that $S(n) = 3T(n/2) + n$, we know that $S(n) \geq T(n)$. We can apply the master method to he function $S(n)$: n is $\Theta(n)$, so $S(n)$ is $\Theta(n^{log_2 3})$. But, as $T(n) \leq S(n)$, we can conclude that $T(n)$ is $O(n^{log_2 3})$. Here we can use only O and not Θ as the function $S(n) \geq T(n)$; As we were only able apply the master method indirectly, we could show a tight bound if we use the result from Akra-Bazzi computed above and compare former [(3T(n/2)] and latter [n/logn] parts we get,

Answer: Upper bound: $n^{log_2 3}$, Lower Bound: $(n(loglogn))$

(b) $T(n) = \sqrt{n}T(\sqrt{n}) + n$

Solution: By Rolling-Unrolling Method.

$T(n) = n^{1/2}T(n^{1/2}) + cn$

$= n^{1/2}\left[n^{1/4}T\left(n^{1/2}\right) + cn^{1/2}\right] + cn$

$= n^{1/2+1/4}T\left(n^{1/2}\right) + cn^{1/2+1/2} + cn$

$= n^{1/2+1/4}T\left(n^{1/8}\right) + cn^{1/4} + 2cn$

$= n^{1/2+1/4+1/8}T\left(n^{1/8}\right) + 3cn$

...

$= n^(1 - 1/2^k)T(n^{1/2^k}) + kcn$

$=$ ... let $k =$log log n so $2^k =$ log n and $n^{2-k} = n^{1/logn} = 2$...

$= n/2T(2) + cn$loglog n $\qquad (O(n/2) < O(n\text{loglog}n))$

$= \Theta(nloglogn)$

Answer: Upper and Lower Bound: $nloglogn$

(c) $T(n) = 3T(n-1)$
Solution: By Rolling-Unrolling Method.
Assuming base case $T(0)$ = constant c. $T(n) = 3T(n-1)$ (Equation 1)
Substituting T(n) with T(n-1) in Equation 1
$= 3[3T((n-1)-1)] = 3^2T(n-2)$
$= 3[3^2T((n-2)-1)] = 3^3T(n-3)$
.
.
$= 3^kT(n-k)$
Let $k = n \rightarrow 3^nT(0)$
Since $T(0)$ is a constant by our assumption, so the complexity of the above recurrence relation turns out to be $\Theta(3^n)$.
Answer: Upper Bound and Lower Bound: 3^n

2 Part 2

Group Hackerrank Name: singhay

1. Surekha Jadhwani : surekha@ccs.neu.edu / jadhwani.s@husky.neu.edu

2. Akash Singh : singhaka@ccs.neu.edu / singh.aka@husky.neu.edu

3. Ayush K. Singh : singhay@ccs.neu.edu / singh.ay@husky.neu.edu