

Sample problems - Prelim 2 - Math 1350 - Fall 2010

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>	<i>K</i>	<i>L</i>	<i>M</i>
0	1	2	3	4	5	6	7	8	9	10	11	12
<i>N</i>	<i>O</i>	<i>P</i>	<i>Q</i>	<i>R</i>	<i>S</i>	<i>T</i>	<i>U</i>	<i>V</i>	<i>W</i>	<i>X</i>	<i>Y</i>	<i>Z</i>
13	14	15	16	17	18	19	20	21	22	23	24	25

1. Write 69 (base 10) in base 3 Solution:

$3^0 = 1, 3^1 = 3, 3^2 = 9, 3^3 = 27, 3^4 = 81$, so we do not need anything bigger than 3^3 . Since $69 \div 27 = 2$ with remainder 15, $15 \div 9 = 1$ with remainder 6 and $6 \div 3 = 2$ with remainder 0, the answer is

$$2120$$

2. Write 1001101 (base 2) in base 10.

Solution: 1001101 base 2 equals

$$\begin{aligned} 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ = 64 + 8 + 4 + 1 = 77. \end{aligned}$$

3. Compute the sum GILACXA + BCADAFH in base 26.

Solution: HKLDDCH

Note: Changing to base 10 is completely impractical during the exam when you can not use a calculator.

4. The following table gives a Boolean function f with 3-bit inputs and 1-bit outputs. Give an explicit formula for f , and simplify as much as possible.

$x_1x_2x_3$	$f(x_1x_2x_3)$
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	0

Solution: $f(x_1x_2x_3) = f_1(x_1x_2x_3) + f_2(x_1x_2x_3) + f_3(x_1x_2x_3) \text{ MOD } 2$, where

$$f_1(x_1x_2x_3) = (1 + x_1) \cdot (1 + x_2) \cdot x_3 \text{ MOD } 2.$$

$$f_2(x_1x_2x_3) = (1 + x_1) \cdot x_2 \cdot x_3 \text{ MOD } 2.$$

$$f_3(x_1x_2x_3) = x_1 \cdot (1 + x_2) \cdot (1 + x_3) \text{ MOD } 2.$$

These three functions simplify to

$$f_1(x_1x_2x_3) = (1 + x_1 + x_2 + x_1 \cdot x_2) \cdot x_3 \text{ MOD } 2.$$

$$f_2(x_1x_2x_3) = x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3 \text{ MOD } 2.$$

$$f_3(x_1x_2x_3) = x_1 \cdot (1 + x_2 + x_3 + x_2 \cdot x_3) \text{ MOD } 2.$$

which is

$$f_1(x_1x_2x_3) = x_3 + x_1 \cdot x_3 + x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3 \text{ MOD } 2.$$

$$f_2(x_1x_2x_3) = x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3 \text{ MOD } 2.$$

$$f_3(x_1x_2x_3) = x_1 + x_1 \cdot x_2 + x_1 \cdot x_3 + x_1 \cdot x_2 \cdot x_3 \text{ MOD } 2.$$

Collecting like terms leaves

$$f(x_1x_2x_3) = x_1 + x_3 + x_1 \cdot x_2 + x_1 \cdot x_2 \cdot x_3 \text{ MOD } 2.$$

5. Show that $\mathcal{O}(n!) \subseteq \mathcal{O}(n^n)$.

Solution: The problem is equivalent to showing that if $f(n) \in \mathcal{O}(n!)$, then $f(n) \in \mathcal{O}(n^n)$. Suppose that $f(n) \in \mathcal{O}(n!)$. By definition there are nonnegative numbers M and N so that $f(n) \leq M \cdot n!$ for all $n \geq N$. Now, to show that $f(n) \in \mathcal{O}(n^n)$ we must find M' and N' so that $f(n) \leq M' \cdot n^n$ for all $n \geq N'$. Since $n! \leq n^n$ for all $n \geq 1$, setting $M' = M$ and $N' = N$ works.

6. Let $\mu(n) = n^2 + 16n$. Is $\mu(n) \in \mathcal{O}(n^3)$?

Solution: Yes. There are several different ways of doing this problem. One is to use $M = 16$ and $N = 2$ in the definition of $\mathcal{O}(n^3)$. Since for all $n \geq 2$, $n(n+1) \leq n^2$ and hence

$$n^2 + 16n \leq 16n^2 + 16n = 16(n^2 + n) = 16n(n+1) \leq 16n^3$$

whenever $n \geq 2$. Another possibility is to use $M = 1$ and $N = 16$ since whenever $n \geq 16$,

$$n^2 + 16n = n(n+16) \leq n(n^2) = n^3$$

Smaller N could also work, with $N = 5$ the smallest possible, but it is easier to see that for $n \geq 16$, $n+16 \leq n^2$.

7. Show that if $\mu(n) \in \mathcal{O}(f(n))$, then $\mathcal{O}(\mu(n))$ is contained in $\mathcal{O}(f(n))$.

Solution: By definition, we must show that if there exists nonnegative M and N such that $\mu(n) \leq Mf(n)$ for all $n \geq N$, then $\mathcal{O}(\mu(n))$ is contained in $\mathcal{O}(f(n))$. This is the same as showing that if there exists nonnegative M and N such that $\mu(n) \leq Mf(n)$ for all $n \geq N$ and $g(n) \in \mathcal{O}(f(n))$, then $g(n) \in \mathcal{O}(\mu(n))$. By definition this is the same as showing that if there exists nonnegative M and N such that $f(n) \leq M\mu(n)$ for all $n \geq N$ and there exist nonnegative M' and N' with $g(n) \leq M'f(n)$ for all $n \geq N'$, then there exist nonnegative M'' and N'' such that $g(n) \leq M''\mu(n)$ for all $n \geq N''$.

Claim: M'' equal to $M \cdot M'$, and N'' equal to the maximum of N and N' works. In fact, if $n \geq N''$, then $n \geq N'$, and $n \geq N$. So

$$g(n) \leq M'f(n) \leq M' \cdot (M\mu(n)) = M''\mu(n).$$

Yes - this is the hardest possible big Oh problem we could ever ask!

8. A linear feedback shift register uses constants $c_1 = 1$, $c_2 = 0$, $c_3 = 1$, and $c_4 = 0$, and has the output stream start with 0001. Use the output as a binary Vigenère key to decipher the cipher text 101100110110.

Solution: The output of the LFSR is 000101000101, so the plain text is

$$101100110110 \oplus 000101000101 = 101001110011.$$

9. The plain text of a binary Vigenère message is 1011011101 and the corresponding cipher text is 1101001100. The key is the output of a 4-bit linear feedback shift register. Determine the constants c_1, c_2, c_3 and c_4 used by the LFSR.

Solution: The key produced by the LFSR is plain text \oplus cipher text which is 0110010001. Starting with the fifth bit we get the following equations.

$$\begin{array}{rclcl} 0 & = & & c_2 + c_3 & \text{MOD } 2 \\ 1 & = & c_1 + & c_2 & \text{MOD } 2 \\ 0 & = & c_1 + & & c_4 \text{ MOD } 2 \\ 0 & = & & c_3 & \text{MOD } 2 \end{array}$$

The last equation says that $c_3 = 0$. Now the first equation tells us that $c_2 = 0$. So the second equation says that $c_1 = 1$ and finally the third equation tells us that $c_4 = 1$.

10. Let $f_k(x)$ be a Boolean function defined by

$$f_{k_1 k_2}(x_1 x_2) = (x_1 + k_1 \cdot x_2 \text{ MOD } 2)(k_2 + x_1 \cdot x_2 \text{ MOD } 2).$$

Suppose the output of the Feistel function $F_{10}(x)$ is 1001. What 4-bit input was x ? Solution: We are asked to find $F_k^{-1}(1001)$. This is

$$\begin{array}{rcll}
R(1001) & \oplus & f_k(L(1001)) & \parallel L(1001) \\
= 01 & \oplus & f_{10}(10) & \parallel 10 \\
= 01 & \oplus & ((1 + 1 \cdot 0 \text{ MOD } 2)(0 + 1 \cdot 0 \text{ MOD } 2)) & \parallel 10 \\
= 01 & \oplus & 10 & \parallel 10 \\
= 1110. & & &
\end{array}$$

11. A hash function $H(x)$ outputs 80-bit binary strings. Suppose that x_1, x_2, \dots, x_8 are eight distinct inputs chosen at random. Assume that H is constructed so that for inputs chosen at random, each output is equally likely. What is the probability that the hash values $H(x_1), H(x_2), \dots, H(x_8)$ are all different? (Do NOT attempt to simplify your answer.)

Solution: The most important observation is that there are 2^{80} possible hashes for one input. The second is that this is just like the birthday problem of what is the probability that 8 different people have distinct birthdays, except that there are 2^{80} possible birthdays! If we follow example 2.6.10 from the text the probability must be

$$\frac{P(2^{80}, 8)}{(2^{80})^8} = \frac{P(2^{80}, 8)}{2^{640}}.$$

Another way to do the problem is as follows. What is the probability that $H(x_1)$ and $H(x_2)$ are different? This is 1- prob. that they are the same which equals $(1 - \frac{1}{2^{80}})$. What about $H(x_3)$ is not equal to $H(x_1)$ and $H(x_2)$? The same reasoning says this is $1 - \frac{2}{2^{80}}$. Since each of these appears to be independent of the previous we get a different looking (but identical) answer of

$$(1 - \frac{1}{2^{80}})(1 - \frac{2}{2^{80}})(1 - \frac{3}{2^{80}})(1 - \frac{4}{2^{80}})(1 - \frac{5}{2^{80}})(1 - \frac{6}{2^{80}})(1 - \frac{7}{2^{80}}).$$