CSE 20 Spring 2022
Homework 2 solutions
Due date: Wednesday, April 13, 2021 at 11:59pm

In this assignment,

You will consider multiple number representations and how they connect to applications in computer science. You will also practice tracing and working with algorithms.

In this class, unless the instructions explicitly say otherwise, you are required to justify all your answers.

1. Let D be the set of all odd length binary strings that start with 1.

   It is defined recursively in the following way:

   Basis Step:

   - $1 \in D$

   Recursive Steps:

   If $x \in D$ then

   - $x00 \in D$
   - $x01 \in D$
   - $x10 \in D$
   - $x11 \in D$

   (a) Consider the function $f : D \to \mathbb{Z}^+$ that takes an odd length string u and returns the value $(u)_2$ converted to a positive integer.

   Describe the function f using a recursive definition.

   Solution:
   Basis Step: $f(1) = 1$
   Recursive step: Let x be an arbitrary element of D. Then:

   - $f(x00) = 4f(x)$
   - $f(x01) = 4f(x) + 1$
   - $f(x10) = 4f(x) + 2$
   - $f(x11) = 4f(x) + 3$

   (b) Consider the function $p : D \to \{0, 1, 2, 3\}^*$ that takes an odd length string u and returns the base-4 expansion of u.

   Describe the function p using a recursive definition.

   (Recall that $\{0, 1, 2, 3\}^*$ is the set of all strings over the alphabet $\{0, 1, 2, 3\}$.

   Solution:
   Basis Step: $p(1) = 1$
   Recursive step: Let x be an arbitrary element of D. Then:

   - $p(x00) = p(x) \circ 0$
   - $p(x01) = p(x) \circ 1$
   - $p(x10) = p(x) \circ 2$
   - $p(x11) = p(x) \circ 3$

2. Is there an advantage to use a different number system that is different than base 10? What number base would be the best? Some say base 12 makes the most sense. In fact, there is at least one organization whose purpose is to educate and advocate for society to adopt a base 12 number system: The Dozenal Society of America.

For this problem, the allowable digits in decimal are $\{0,1,2,3,4,5,6,7,8,9\}$ and the allowable numerals for dozenal are $\{\mathit{0,1,2,3,4,5,6,7,8,9,X,E}\}$ (Notice that the dozenal numerals are all in italics so that we know which number system we should be using.)

The numeral *X* is called "dek" which is equal to the number 10 in decimal and the numeral *E* is called "el" which is equal to the number 11 in decimal.

The number *10* is called "do" which is short for "dozen" which is equal to 12 in decimal. The number *100* is called "gro" which is short for "gross" which is equal to 144 in decimal. Multiples of *10* like *20* for example is called "two-do".

For example, the number of stars in this string: $*********************$ can be described as $21 = (21)_{10}$ (in decimal) or by $\mathit{19} = (19)_{12}$ (in dozenal).

One reason why dozenal is preferred over decimal is because the proper divisors of $12 = \mathit{10}$ are $\{\mathit{1,2,3,4,6}\}$ and the proper divisors of $10{=}X$ are $\{1,2,5\}$. With more proper divisors, some aspects of arithmetic in dozenal is easier than in decimal.

Exercise 1: (use any method. please show your work.)

Convert each of the decimal numbers into dozenal: (show your work by tracing one of the two algorithms from class.)

- 2022

  Solution:

q	k	$q \neq 0$	a_k
2022	0	yes	$a_0 = 2022$ **mod** $12 = 6$
2022 **div** $12 = 168$	1	yes	$a_1 = 168$ **mod** $12 = 0$
168 **div** $12 = 14$	2	yes	$a_2 = 14$ **mod** $12 = 2$
14 **div** $12 = 1$	3	yes	$a_3 = 1$ **mod** $12 = 1$
1 **div** $12 = 0$	4	no	

  So $2022 = \mathit{1206}$

- 515

  Solution:

q	k	$q \neq 0$	a_k
515	0	yes	$a_0 = 515$ **mod** $12 = 11$
515 **div** $12 = 42$	1	yes	$a_1 = 42$ **mod** $12 = 6$
42 **div** $12 = 3$	2	yes	$a_2 = 3$ **mod** $12 = 3$
3 **div** $12 = 0$	3	no	

  So $515 = \mathit{36E}$

Convert each of the dozenal numbers into decimal:

- *XEXE*

  Solution:

  $10 * 12^3 + 11 * 12^2 + 10 * 12 + 11 = 18995$

- *2022*

  Solution:

  $2 * 12^3 + 0 * 12^2 + 2 * 12 + 2 = 3482$

Since dozenal is a positional number system, arithmetic can be done in the same manner as we learned in gradeschool.

Exercise 2:

Perform gradeschool addition on the two dozenal numbers: (you can upload a handwritten answer for this problem.)

$$\begin{array}{cccccc} & X & 4 & 1 & 9 & 7 \\ + & 8 & E & X & X & X \\ \hline \end{array}$$

Solution:

$$
\begin{array}{ccccccc}
 & \scriptstyle 1 & \scriptstyle 1 & \scriptstyle 1 & \scriptstyle 1 & \scriptstyle 1 & \\
 & & X & 4 & 1 & 9 & 7 \\
+ & & 8 & E & X & X & X \\
\hline
 & 1 & 7 & 4 & 0 & 8 & 5
\end{array}
$$

Exercise 3:

Perform gradeschool multiplication on the two dozenal numbers: (you can upload a handwritten answer for this problem.)

$$
\begin{array}{ccc}
 & X & 7 \\
\times & 9 & 5 \\
\hline
\end{array}
$$

Solution:

$$
\begin{array}{ccccc}
 & & \scriptstyle 7 & \scriptstyle 5 & \\
 & & \scriptstyle 4 & \scriptstyle 2 & \\
 & & & X & 7 \\
\times & & & 9 & 5 \\
\hline
 & \scriptstyle 1 & & & \\
 & & 4 & 4 & E \\
+ & 7 & E & 3 & 0 \\
\hline
 & 8 & 3 & 7 & E
\end{array}
$$

Another benefit of dozenal is when you use it to represent fractional quantities.

The analogous notation of a "decimal point" is a "dozenal semicolon". The way it works is similar to a decimal point.

For example, the number 0.1 is equal to the fraction 1/10 and the number *0;1* is equal to the fraction *1/10* $= 1/12$. Similarly, *0;01=1/100* $= 1/144$ and *0;001=1/1000* $= 1/1728$.

Exercise 4:

For each decimal fraction, write the dozenal semicolon notation of the number. For example, for the fraction 1/2, the dozenal semicolon notation would be *0;6* since $1/2 = 6/12$.

- 5/9

  Solution:
  $5/9 = 6/12 + 8/144 =$ *0;68*
- 11/16

  Solution:
  $11/16 = 8/12 + 3/144 =$ *0;83*
- 12/18

  Solution:
  $12/18 = 8/12 =$ *0;8*

Exercise 5: (for fair effort completeness:)

What is one example (not mentioned in the homework) where you would prefer dozenal over decimal?

What is one example (not mentioned in the homework) where you would prefer decimal over dozenal?

3. The *radix economy* of a positive integer n and a base b is equal to b times the number of numerals it would take in base b to represent n.

   For example, if the number $n = 77675$ and $b = 10$, then we need 5 numerals to represent n then we multiply by $b = 10$ since there are 10 possible digits. So in this example, the radix economy of 77675 in base 10 is $5 * 10 = 50$.

Let's look at this same number $n = 77675$ with base $b = 6$. Then $77675 = (1355335)_6$ requires 7 numerals. So the radix economy in this case is $7 * 6 = 42$ which is lower than 50.

The idea is that for higher bases, the number of numerals needed is fewer than for lower bases but higher bases require more numerals to remember.

Exercise 1:

For each base in the set $\{2, 3, 4, 5, 6, 7, 8, 9\}$, compute the radix economy of 77675.

Which base gives the lowest radix economy?

Solution:

- In base 2, you would need: 17 bits.

$$2^{16} = 65536 \leq 77675 < 2^{17} = 131072$$

  so the radix economy is $2 * 17 = 34$

- In base 3, you would need: 11 numerals.

$$3^{10} = 59049 \leq 77675 < 3^{11} = 177147$$

  so the radix economy is $3 * 11 = 33$

- In base 4, you would need: 9 numerals.

$$4^8 = 65536 \leq 77675 < 4^9 = 262144$$

  so the radix economy is $4 * 9 = 36$

- In base 5, you would need: 7 numerals.

$$5^6 = 15625 \leq 77675 < 5^7 = 78125$$

  so the radix economy is $5 * 7 = 35$

- In base 6, you would need: 7 numerals.

$$6^6 = 46656 \leq 77675 < 6^7 = 279936$$

  so the radix economy is $6 * 7 = 42$

- In base 7, you would need: 6 numerals.

$$7^5 = 16807 \leq 77675 < 7^6 = 117649$$

  so the radix economy is $7 * 6 = 42$

- In base 8, you would need: 6 numerals.

$$8^5 = 32768 \leq 77675 < 8^6 = 262144$$

  so the radix economy is $8 * 6 = 48$

- In base 9, you would need: 6 numerals.

$$9^5 = 59049 \leq 77675 < 9^6 = 531441$$

  so the radix economy is $9 * 6 = 54$

Exercise 2: In general, for some n and some base b, then explain why the function f such that $f(n, b)$ is the radix economy of n with base b:

$f : \mathbb{Z}^+ \times \mathbb{Z}^{\geq 2} \to \mathbb{N}$

$$f(n, b) = b\lfloor \log_b(n) + 1 \rfloor$$

Solution:
$\lfloor \log_b(n) + 1 \rfloor$ is the minimum number of numerals needed to represent n in base b. So since the radix economy is the product of b and the minimum number of numerals needed to represent n in base b, $f(n, b) = b\lfloor \log_b(n) + 1 \rfloor$.
(The reason that $\lfloor \log_b(n) + 1 \rfloor$ is the minimum number of numerals required is because the number $b^{k-1} = (1000\ldots00)_b$ is the smallest number that requires k numerals and $b^k - 1 = (BBB\ldots BBB)_b$ is the largest number that requires k numerals (with B is the numeral representing $b-1$.) So, assuming that n requires k numerals, then $b^{k-1} \leq n < b^k$. Take the log of all parts to get:

$$(k-1)\log_b(b) \leq \log_b(n) < \log_b(b^k)$$
$$k - 1 \leq \log_b(n) < k$$

Then add 1 to all parts to get:

$$k \leq \log_b(n) + 1 < k + 1$$

so, when we take the floor of $\log_b(n) + 1$, then we will get k which is the number of numeral required.)

Exercise 3: (for fair effort completeness:)
Is there a particular base that will always give the optimal radix economy? How does this relate to computer design? Is base 2 the "best" base to be using? why or why not?

Solution:
A continuous approximation of $f(n, b)$ is $b\log_b(n)$. Let's fix n and try to minimize b by taking derivatives:

$$F(b) = b\log_b(n) = \frac{b\ln(n)}{\ln(b)}$$
$$\frac{d}{db}F(b) = \frac{\ln(n) * \ln(b) - b\ln(n)(1/b)}{\ln(b)^2}$$

Set this equal to 0 and solve for b:

$$\frac{\ln(n) * \ln(b) - b\ln(n)(1/b)}{\ln(b)^2} = 0$$
$$\ln(n) * \ln(b) - \ln(n) = 0$$
$$\ln(b) - 1 = 0$$
$$\ln(b) = 1$$
$$b = e$$

So, using the continuous approximation, the minimum occurs when $b = e \approx 2.72$. So, since 3 is the closest whole number to 2.72, it makes sense that 3 should have the lowest radix economy. This is true on average but there exist numbers where 2 (and 4) has a better radix economy. For example, if $n = 250$ then in base 2, you need 8 bits so the radix economy is $2 * 8 = 16$ but in base 3, you need 6 numerals so the radix economy is $3 * 6 = 18$. (Note that in base 4, you need 4 numerals to get the same radix economy as in base 2: $4 * 4 = 16$.)

4. (**20 points**) Color in computer is often represented as a 3-tuple (R, G, B) such that R, G, B are each an integer ranging from 0 to 255.

The R value is the red component, the G value green component and B, the blue component. For example: $(0, 0, 0)$ represents black and $(255, 255, 255)$ represents white.

We can use hexadecimal (base 16) to represent each color. Recall that the allowable "numerals" for base 16 are: $H = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$. Then since each component is a value between 0 and 255, we can use two hexadecimal numerals for each component. (This is because with two hexadecimal numerals, you can represent all integers between $(00)_{16,2} = 0$ and $(FF)_{16,2} = 255$.)

A **hex color** is an integer n that has a base 16 *fixed-width* 6 expansion:

$$n = (r_1r_2g_1g_2b_1b_2)_{16,6}$$

where $(r_1r_2)_{16,2}$ is the red component, $(g_1g_2)_{16,2}$, the green component, and $(b_1b_2)_{16,2}$, the blue component.

(a) What is the red, green, and blue values each in base 10 of the hex color: $(82C09F)_{16,6}$

Solution:

- Red: $8 * 16 + 2 = 130$
- Green: $12 * 16 + 0 = 192$
- Blue: $9 * 16 + 15 = 159$

(b) What is the **hex color** that corresponds to the (R, G, B) values of $(65, 173, 211)$?

Solution:

- Red: $65 = 4 * 16 + 1 = (41)_{16}$
- Green: $173 = 10 * 16 + 13 = (AD)_{16}$
- Blue: $211 = 13 * 16 + 3 = (D3)_{16}$

$(41ADD3)_{16,6}$

(c) A color is considered "gray" if the red, green and blue components are all equal.

Often, we would like to "desaturate" a color by converting to a gray color.

One way to do this is to calculate a gray value and repeat the value each for the red, green and blue components. A function to compute the gray value is described below:

(Recall that for this problem, $H = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.)

Consider the function G:

- **Domain:** the set of all fixed width **6** hexadecimal integers.
- **Codomain:** the set of all integers in the range 0 to 255.
- **Rule:** for any input $(r_1r_2g_1g_2b_1b_2)_{16,6}$,

$$G((r_1r_2g_1g_2b_1b_2)_{16,6}) = \lfloor ((r_1r_2)_{16} + (g_1g_2)_{16} + (b_1b_2)_{16})/3 \rfloor$$

i. Evaluate $G((E351C0)_{16})$. (Show your work. It is your choice to have your answer in base 16 or base 10).

Solution:

- $(E3)_{16} = 14 * 16 + 3 = 227$
- $(51)_{16} = 5 * 16 + 1 = 81$
- $(C0)_{16} = 12 * 16 + 0 = 192$

$$\lfloor (227 + 81 + 192)/3 \rfloor = 166$$

ii. What is the output of the function G if the input is the hex color that corresponds to the (base 10) integer 202362? (Please show your work.)

q	k	$q \neq 0$	a_k
202362	0	yes	$a_0 = 202362$ **mod** $16 = 10$
202362 **div** $16 = 12647$	1	yes	$a_1 = 12647$ **mod** $12 = 7$
12647 **div** $16 = 790$	2	yes	$a_2 = 790$ **mod** $16 = 6$
790 **div** $16 = 49$	3	yes	$a_3 = 49$ **mod** $16 = 1$
49 **div** $16 = 3$	4	yes	$a_4 = 3$ **mod** $16 = 3$
3 **div** $16 = 0$	5	no	

So $202362 = (3167A)_{16} = (03167A)_{16,6}$

Then:

- $(03)_{16} = 0 * 16 + 3 = 3$
- $(16)_{16} = 1 * 16 + 6 = 22$
- $(7A)_{16} = 7 * 16 + 10 = 122$

Then

$$\lfloor (3 + 22 + 122)/3 \rfloor = 49$$

iii. (for fair effort completeness)
What happens if you input a gray color (a color that has equal red, green and blue components)? Is it what you expect to happen?

5. A standard way for computers to represent non-negative and negative integers x using binary fixed-width N is the *2's complement* method $T_N(x)$.

   - Non-negative integers x with $0 \leq x \leq 2^{N-1} - 1$ are represented using ordinary fixed-width binary (e.g. Using 2's complement fixed-width 8, $T_8(30) = (00011110)_{2c,8}$) because $(11110)_2$ is equal to 30.
   - Negative integers x with $-2^{N-1} \leq x < 0$ are represented by converting the sum $x + 2^N$ into fixed width 8 binary.

   $$T_8(-30) = (11100010)_{2c,8}$$

   Because $2^N + (-30) = 2^8 - 30 = 226$ and $(11100010)_2$ is equal to 226.

   (a) Convert each base 10 integer to *2's complement* fixed-length 8: (no justification necessary)

   - 88

     Solution:
     $(0101\ 1000)_{2c,8}$
   - -99

     Solution:
     $(1001\ 1101)_{2c,8}$

   (b) Convert each *2's complement* fixed-length 8 integer into a base 10 integer. (show your work on how you did the conversion.)

   - $(10001100)_{2c,8}$

     Solution:
     Since 1 is the first bit, we must subtract the remaining bits from $2^7 = 128$. $(0001100)_{2,7} = 12$, $128 - 12 = 116$.
     So $(10001100)_{2c,8} = -116$
   - $(11100111)_{2c,8}$

     Solution:
     Since 1 is the first bit, we must subtract the remaining bits from $2^7 = 128$. $(1100111)_{2,7} = 103$, $128 - 103 = 25$.
     So $(11100111)_{2c,8} = -25$

   (c) One of the great things about *2's complement* is that you can add positive and negative integers together using regular binary addition. The trick is to ignore the overflow.

   Consider the two numbers $x = (10011101)_{2c,8}$ and $y = (01001110)_{2c,8}$.

   Write out the gradeschool addition of the x and y and ignore the overflow.

   Convert x, y and the result of the addition all into base 10 to check if the result is correct. *you can neatly handwrite this part if you wish.*

Solution:

$$\begin{array}{ccccccccc} & & & 1 & 1 & 1 & & & \\ & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ + & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{array}$$

$x = -99$, $y = 78$, $x + y = -21$ which is what it should be!!

(d) (for fair effort completeness)

Compare the work with using sign-magnitude:

Consider the two numbers $x = (11100011)_{s,8}$ and $y = (01001110)_{s,8}$.

How would you compute $x + y$ for this example and in general?

Which do you like better? two's complement or sign-magnitude?

Solution:

In order to add two integers with different signs using sign magnitude, you would have to subtract one magnitude from the other (depending on which magnitude is greater.)

I like 2's complement better since you do not have to introduce another arithmetic algorithm. You can just add the binary strings as they are.