HW2 Numbers

CSE20F21

Sample Solutions

Assigned questions

1. (*Graded for fair effort completeness*[1]) Pick a nonnegative integer between 100 and 1000 (inclusive) and express it using at least three different representations, at least two of which must be ones discussed in class. Include at least one trace of using procedure $baseb1$ from page 16 of the Week 0 and 1 notes to calculate the base b_1 expansion of your number (for your choice of b_1) and include at least one trace of using procedure $baseb2$ from page 16 of the Week 0 and 1 notes to calculate the base b_2 expansion of your number (for your choice of b_2).

   Extra; not for credit: Consider choosing representations that might be useful in some way; what would make one representation more useful than another?

   Solution: Let's consider the nonnegative integer 100. We'll represent it using octal representation, hexadecimal representation, and binary fixed-width 16 representation.

   - To find the octal representation, we use $baseb1$. Let's trace the procedure with $n = 100, b = 8$:
     - In lines 2-3, the variables v and k are initialized to $v := 100$, $k := logb(100, 8) + 1$ and since $8^2 = 64 \leq 100 < 512 = 8^3$, $logb(100, 8) = 2$ and $k := 3$.
     - In line 4, we enter the for loop, iterating as i goes from 1 to 3.
     - When $i := 1$, a_2 is initialized to 0 in line 5, and then updated to 1 in line 7 because $100 \geq 8^2$, and v is updated to $100 - 64 = 36$ in line 8. We exit the while loop because $36 \not\geq 8^2$.
     - When $i := 2$, a_1 is initialized to 0 in line 5, and then updated to 1 in line 7 because $36 \geq 8^1$, and v is updated to $36 - 8 = 28$. We iterate through the while loop again because $28 \geq 8^1$, so a_1 is updated to 2 in line 7 and v is updated to $28 - 8 = 20$ in line 8. We iterate through the while loop again because $20 \geq 8^1$, so a_1 is updated

[1] This means you will get full credit so long as your submission demonstrates honest effort to answer the question. You will not be penalized for incorrect answers.

to 3 in line 7 and v is updated to $20 - 8 = 12$ in line 8. We iterate through the while loop again because $12 \geq 8^1$, so a_1 is updated to 4 in line 7 and v is updated to $12 - 8 = 4$ in line 8. We exit the while loop because $4 \not\geq 8^1$.

  - When $i := 3$, a_0 is initialized to 0 in line 5, and then updated to 1 in line 7 because $4 \geq 8^0$, and v is updated to $4 - 1 = 3$ in line 8. We iterate through the while loop again because $3 \geq 8^0$, so a_0 is updated to 2 in line 7 and v is updated to $3 - 1 = 2$ in line 8. We iterate through the while loop again because $2 \geq 8^0$, so a_0 is updated to 3 in line 7 and v is updated to $2 - 1 = 1$ in line 8. We iterate through the while loop again because $1 \geq 8^0$, so a_0 is updated to 4 in line 7 and v is updated to $1 - 1 = 0$ in line 8. We exit the while loop because $0 \not\geq 8^0$.

  The procedure returns the octal expansion $(a_2a_1a_0)_8 = (144)_8$.

  Note: We can check our work by translating back to decimal using the definition of octal expansion:

$$(144)_8 = 1 \cdot 8^2 + 4 \cdot 8^1 + 4 \cdot 8^0 = 64 + 32 + 4 = 100$$

  as expected.

- To find the hexadecimal representation, we use *baseb*2. Let's trace the procedure with $n = 100, b = 16$:
  - In lines 2-3, the variables q and k are initialized to $q := 100$ and $k := 0$.
  - We enter the while loop in line 4 because $100 \neq 0$.
  - Since $100 = 6 \cdot 16 + 4$, 100 **div** $16 = 6$ and 100 **mod** $16 = 4$. Thus, in line 5 a_0 gets the value 4, q is updated to the value 6 in line 6, and the value of k is incremented to 1.
  - Since $6 = 0 \cdot 16 + 6$, in line 5 a_1 gets the value 6, q is updated to the value 0 in line 6, and the value of k is incremented to 2.

  The procedure returns the hexadecimal expansion $(64)_{16}$.

  Note: We can check our work again by translating back to decimal using the definition of hexadecimal expansion:

$$(64)_{16} = 6 \cdot 16^1 + 4 \cdot 16^0 = 96 + 4 = 100$$

  as expected.

- To find the binary fixed-width 16 representation, we first express 100 as a sum of powers of 2 (to get the binary expansion) and then pad with leading zeros.

$$100 = 64 + 32 + 4 = 1 \cdot 2^6 + 1 \cdot 2^4 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = (1100100)_2$$

  Since the binary expansion has 7 columns, we need to add 9 leading zeros to get

$$100 = (0000000001100100)_{2,16}$$

2. In this question, we will use recursive definitions to give a precise description of the set of strings that form octal (base 8) expansions and the values of those expansions. Let's start by defining the set of possible coefficients

$$C_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$$

(a) (*Graded for fair effort completeness*) Fill in the recursive definition of the set of all strings that form octal expansions, S_8:

Definition (Solution) The set S_8 is defined (recursively) by:

Basis Step:	If $x \in \underline{\{1, 2, 3, 4, 5, 6, 7\}}$, then $x \in S_8$
Recursive Step:	If $s \in S_8$ and $x \in C_8$, then $\underline{sx} \in S_8$

where notice that the basis step doesn't allow 0 because we can't have leading zeros in base b expansions and notice that sx in the recursive step is the result of string concatenation.

(b) (*Graded for correctness*[2]) Consider the function $v_8 : S_8 \to \mathbb{Z}^+$ defined recursively by

Basis Step: If $x \in C_8$, then $v_8(x) = x$.

Recursive Step: If $s \in S_8$ and $x \in C_8$, then $v_8(sx) = 8v_8(s) + x$, where the input sx is the result of string concatenation and the output $8v_8(s)$ is the result of integer multiplication.

Calculate $v_8(104)$, including all steps in your calculation and justifications for them.

Solution:

$$\begin{aligned} v_8(104) &= 8v_8(10) + 4 && \text{By the recursive step, with } s = 10 \text{ and } x = 4 \\ &= 8\,(8v_8(1) + 0) + 4 && \text{By the recursive step, with } s = 1 \text{ and } x = 0 \\ &= 8\,(8 \cdot 1 + 0) + 4 && \text{By the basis step, with } x = 1 \\ &= 64 + 4 = 68 \end{aligned}$$

Notice that this is the value of the octal expansion $(104)_8$.

(c) (*Graded for fair effort completeness*) It turns out[3] that for any string u in S_8, the value of the octal expansion $(u)_8$ equals $v_8(u)$. Using this fact, write an expression relating the value of $(u000)_8$ to the value of $(u)_8$ and justify it.

Solution: Given that $(u000)_8 = v_8(u000)$ and $(u)_8 = v_8(u)$, we can use the recursive definition of v_8 to relate these two values:

$$\begin{aligned} v_8(u000) &= 8v_8(u00) + 0 && \text{By the recursive step, with } s = u00 \text{ and } x = 0 \\ &= 8\,(8v_8(u0) + 0) + 0 && \text{By the recursive step, with } s = u0 \text{ and } x = 0 \\ &= 8\,(8\,(8v_8(u) + 0) + 0) + 0 && \text{By the recursive step, with } s = u \text{ and } x = 0 \\ &= 8^3 v_8(u) \end{aligned}$$

[2] This means your solution will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

[3] We'll be able to prove this in Week 7 or so, once we've talked about induction.

Thus, $(u000)_8 = 8^3(u)_8$ and we see that shifting the octal expansion to the left by adding three trailing zeros results in multiplying the value by 8^3.

3. (*Graded for correctness*) Recall that, mathematically, a color can be represented as a 3-tuple (r, g, b) where r represents the red component, g the green component, b the blue component and where each of r, g, b must be from the collection $\{x \in \mathbb{N} \mid 0 \leq x \leq 255\}$.

   As an **alternative** representation, in this assignment we'll use base 16 fixed-width expansions to represent colors as single numbers.

   Definition: A **hex color** is a nonnegative integer less than or equal to 16777215. For n a hex color, we define its red, green, and blue components by first writing its base 16 fixed-width 6 expansion

   $$n = (r_1 r_2 g_1 g_2 b_1 b_2)_{16,6}$$

   and then defining $(r_1 r_2)_{16,2}$ is the red component, $(g_1 g_2)_{16,2}$ is the green component, and $(b_1 b_2)_{16,2}$ is the blue component.

   Sample response that can be used as reference for the detail expected in your answer:

   In RGB codes[4] white is represented as maximum red, maximum green, and maximum blue and so has $(FF)_{16,2}$ as each of these components. This means that the hex color for white is $(FFFFFF)_{16,6}$ which is the value

   $$15 \cdot 16^5 + 15 \cdot 16^4 + 15 \cdot 16^3 + 15 \cdot 16^2 + 15 \cdot 16^1 + 15 \cdot 16^0 = 16^6 - 1 = 16777215$$

   (a) Write the hex color representing red (with no green or blue) in base 16 fixed-width 6 and also calculate its value (using usual mathematical conventions). Include your (clear, correct, complete) calculations.

   Solution: Red is represented as maximum red, zero green, and zero blue and so has $(FF)_{16,2}$ as the red component and $(00)_{16,2}$ as the other two components. This means that the hex color for red is $(FF0000)_{16,6}$ which is the value

   $$15 \cdot 16^5 + 15 \cdot 16^4 + 0 \cdot 16^3 + 0 \cdot 16^2 + 0 \cdot 16^1 + 0 \cdot 16^0 = 15728640 + 983040 = 16711680$$

   (b) Write the hex color representing green (with no red or blue) in base 16 fixed-width 6 and also calculate its value (using usual mathematical conventions). Include your (clear, correct, complete) calculations.

   Solution: Green is represented as maximum green, zero red, and zero blue and so has $(FF)_{16,2}$ as the green component and $(00)_{16,2}$ as the other two components. This means that the hex color for green is $(00FF00)_{16,6}$ which is the value

   $$0 \cdot 16^5 + 0 \cdot 16^4 + 15 \cdot 16^3 + 15 \cdot 16^2 + 0 \cdot 16^1 + 0 \cdot 16^0 = 61440 + 3840 = 65280$$

[4] You can use online tools to visualize the colors associated with different values for the red, green, and blue components, e.g. `https://www.w3schools.com/colors/colors_rgb.asp`.

(c) Write the hex color representing blue (with no red or green) in base 16 fixed-width 6 and also calculate its values (using usual mathematical conventions). Include your (clear, correct, complete) calculations.

Solution: Blue is represented as maximum blue, zero red, and zero green and so has $(FF)_{16,2}$ as the blue component and $(00)_{16,2}$ as the other two components. This means that the hex color for blue is $(0000FF)_{16,6}$ which is the value

$$0 \cdot 16^5 + 0 \cdot 16^4 + 0 \cdot 16^3 + 0 \cdot 16^2 + 15 \cdot 16^1 + 15 \cdot 16^0 = 240 + 15 = 255$$

(d) The human eye can't distinguish between some hex colors because of physical limitations. Give an example of two hex colors c_1 and c_2 such that they look indistinguishable (the colors they represent are very very similar) but

$$c_1 - c_2 > 50000$$

Justify your choice with (clear, correct, complete) calculations and/or references to definitions, and connecting these calculations and/or definitions with the desired properties. Include squares with each of your two colors so that we can see how indistinguishable they are.

Pro tip: To show a color, you can use the following LaTeX source code:

```
\definecolor{UCSDaccent}{RGB}{0,198,215}
\textcolor{UCSDaccent}{\rule{1cm}{1cm}}
```

which produces

Notice that the code to define the color uses the decimal-like values for each of the red, green, and blue components. For the UCSD accent color we defined, the base 16 fixed-width 2 values are: red is $(00)_{16,2}$, green is $(C6)_{16,2}$, blue is $(D7)_{16,2}$.

Solution: We notice from part (a) that the red component has very high weight in the hex color representation. So, we consider two colors that have just red and whose red values differ by very little. The first color is $c_1 = (FF0000)_{16,6}$ and the second color is $c_2 = (FE0000)_{16,6}$ When we calculate the difference between the colors is

$$\begin{aligned} c_1 - c_2 &= (FF0000)_{16,6} - (FE0000)_{16,6} \\ &= (15 \cdot 16^5 + 15 \cdot 16^4 + 0 \cdot 16^3 + 0 \cdot 16^2 + 15 \cdot 16^1 + 15 \cdot 16^0) \\ &\quad - (15 \cdot 16^5 + 14 \cdot 16^4 + 0 \cdot 16^3 + 0 \cdot 16^2 + 15 \cdot 16^1 + 15 \cdot 16^0) \\ &= 16^4 = 65536 \end{aligned}$$

which is greater than 50000, as required.

The two colors c_1 and c_2 are indistinguishable, as we can see from these pixels

c1 (with RGB(255,0,0) because $(FF)_{16,2} = 255$) and

c2 (with RGB(254,0,0) because $(FE)_{16,2} = 255$):

Extra; not for credit: What does this mean about the choice of hex color for representing colors? What are advantages and disadvantages of this representation?

4. (*Graded for correctness*) In class (Week 2 notes page 7), we discussed fixed-width addition. In this question we will look at fixed-width multiplication. The algorithm for fixed-width multiplication is to multiply using the usual long-multiplication algorithm (column-by-column and carry), and dropping all leftmost columns so the result is the same width as the input terms. For each of the examples below, consider whether this algorithm gives the correct value for the product of the two numbers, based on the way the bitstrings are interpreted.

Sample response that can be used as reference for the detail expected in your answer:

The fixed-width 5 multiplication of $[00101]_{2c,5}$ and $[00101]_{2c,5}$ does not give the correct value for the product, as we can see from the following calculation.

First, we calculate the values:

$$[00101]_{2c,5} = 0 \cdot (-2^4) + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 1 = 5$$

so the correct value for the product is $5 \cdot 5 = 25$, which cannot be represented in 2s complement width 5 (the largest positive number that can be represented in 2s complement width 5 is $[01111]_{2c,5} = 8 + 4 + 2 + 1 = 15$).

When we perform the fixed-width 5 multiplication algorithm:

$$\begin{array}{r}
0\,0\,1\,0\,1 \\
\times 0\,0\,1\,0\,1 \\
\hline
0\,0\,1\,0\,1 \\
+0\,0\,0\,0\,0 \\
+0\,0\,1\,0\,1 \\
\hline
1\,1\,0\,0\,1
\end{array}$$

we get $[11001]_{2c,5} = 1 \cdot (-2^4) + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = -16 + 8 + 1 = -7$, which is not the required value.

(a) Does the fixed-width 5 multiplication of $[11101]_{2c,5}$ and $[11011]_{2c,5}$ give the correct value for the product? Justify your answer with (clear, correct, complete) calculations and/or references to definitions, and connecting these calculations and/or definitions with your answer.

Solution: The fixed-width 5 multiplication of $[11101]_{2c,5}$ and $[11011]_{2c,5}$ does give the correct value for the product, as we can see from the following calculation.

First, we calculate the values:

$$[11101]_{2c,5} = 1 \cdot (-2^4) + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = -16 + 8 + 4 + 1 = -3$$

and

$$[11011]_{2c,5} = 1 \cdot (-2^4) + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = -16 + 8 + 2 + 1 = -5$$

so the correct value for the product is $(-3) \cdot (-5) = 15$, which is represented in 2s complement width 5 as $[01111]_{2c,5} = 8 + 4 + 2 + 1 = 15$.

When we perform the fixed-width 5 multiplication algorithm:

$$\begin{array}{r}
1\ 1\ 1\ 0\ 1 \\
\times 1\ 1\ 0\ 1\ 1 \\
\hline
1\ 1\ 1\ 0\ 1 \\
+1\ 1\ 1\ 0\ 1 \\
+0\ 0\ 0\ 0\ 0 \\
+1\ 1\ 1\ 0\ 1 \\
+1\ 1\ 1\ 0\ 1 \\
\hline
1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1
\end{array}$$

and dropping all leftmost columns to leave just five columns gives 01111, as required.

(b) Does the fixed-width 5 multiplication of $[00100]_{2c,5}$ and $[11100]_{2c,5}$ give the correct value for the product? Justify your answer with (clear, correct, complete) calculations and/or references to definitions, and connecting these calculations and/or definitions with your answer.

Solution: The fixed-width 5 multiplication of $[00100]_{2c,5}$ and $[11100]_{2c,5}$ does give the correct value for the product, as we can see from the following calculation.

First, we calculate the values:

$$[00100]_{2c,5} = 0 \cdot (-2^4) + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 4$$

and

$$[11100]_{2c,5} = 1 \cdot (-2^4) + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = -16 + 8 + 4 = -4$$

so the correct value for the product is $4 \cdot (-4) = -16$, which is represented in 2s complement width 5 as $[10000]_{2c,5}$).

When we perform the fixed-width 5 multiplication algorithm:

$$
\begin{array}{rrrrrrrrrr}
 & & & & & 0 & 0 & 1 & 0 & 0 \\
 & & & & \times & 1 & 1 & 1 & 0 & 0 \\
\cline{6-10}
 & & & & & 0 & 0 & 0 & 0 & 0 \\
 & & & +0 & 0 & 0 & 0 & 0 & & \\
 & & +0 & 0 & 1 & 0 & 0 & & & \\
 & +0 & 0 & 1 & 0 & 0 & & & & \\
+0 & 0 & 1 & 0 & 0 & & & & & \\
\cline{1-10}
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \\
\end{array}
$$

and dropping all leftmost columns to leave just five columns gives 10000, as required.