# IST 4: Planned Schedule - Spring 2015

<table>
<thead>
<tr>
<th>mon</th>
<th>tue</th>
<th>wed</th>
<th>thr</th>
<th>fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>M¹</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>oh</td>
<td>M¹</td>
<td>oh</td>
</tr>
<tr>
<td>13</td>
<td>oh</td>
<td>1</td>
<td>M²</td>
<td>T</td>
</tr>
<tr>
<td>20</td>
<td>oh</td>
<td>oh</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>oh</td>
<td>M²</td>
<td>oh</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>oh</td>
<td></td>
<td>oh</td>
</tr>
<tr>
<td>11</td>
<td>oh</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>oh</td>
<td>oh</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>oh</td>
<td></td>
<td>oh</td>
</tr>
<tr>
<td>1</td>
<td>oh</td>
<td>5</td>
<td></td>
<td>oh</td>
</tr>
</tbody>
</table>

- **T** = today
- **x** = hw#x out
- **Mx** = MQx out
- **Mx** = MQx due
- **oh** = office hours
- **M²** = Midterms
Memory
A word that is associated with the following?
A word that is associated with the following?
Memory and Thought
Me

Information and Thought

and

Logic
“With no effort, he had learned English, French, Portuguese and Latin. I suspect, however, that he was not very capable of thought.”

“To think is to forget differences, generalize, make abstractions. In the teeming world of Funes, there were only details, almost immediate in their presence.”
You are invited to write short essay on the topic of the Magenta Question.

• Recommended length is 3 pages (not more)
• Submit the essay in PDF format to ta4@paradise.caltech.edu
  file name lastname-firstname.pdf

• No collaboration. No extensions

Grading of MQ:
3 points (out of 103)

50% for content quality, 50% for writing quality

Some students will be given an opportunity to give a short presentation for up to 3 additional points
Last year Memory

Thursday, 6/5/2014 2:30pm - MQ2

1. Willis Nguy - A Greater Loss
2. Sarah Brandsen - The Woman Who Cannot Forget
3. Angela Gui - The Persistence of Language
4. Jiyun Ivy Xiao - How I Trained My Memory
5. Ankit Kumar - The Vedic Oral Tradition
6. Leon Ding - Chess Memory
7. Nancy Wen - Scent of a Memory
8. Grace Lee - A Memory about Art
Babylonian Clay Tablets
Greek Proofs...

$\sqrt{2} \ ? = \ ? \frac{p}{q}$

$a^2 + b^2 = c^2$

Memory of mathematical knowledge
\[ \sqrt{2} \quad ? = ? \quad \frac{p}{q} \]
\[ a = 11.7 \]
\[ b = 8.27 \]
\[ \frac{a}{b} = \frac{b}{\frac{a}{2}} \]
\[ \frac{a}{b} = \sqrt{2} \]
A key idea in Information

Is there a finite universal set of building blocks?

Can construct 'everything'

DNA

ABCDE...

123...
A key idea in Information

Is there a finite universal set of building blocks?

Can construct 'everything'

squares
squares
squares
squares
Squaring the Rectangle
Can we find a smaller number of identical squares?
Can we find a smaller number of identical squares?

How?
Idea: Be greedy

How?
We are almost done!

Now what?

From Geometry to Numbers?
Euclidean algorithm for finding the GCD

Greatest Common Divisor

\[
\begin{align*}
40 &= 2 \times 15 + 10 \\
15 &= 1 \times 10 + 5 \\
10 &= 2 \times 5
\end{align*}
\]

Euclid, 300BC
\[ \text{GCD}(561, 208) = 1 \]

\[ 561 = 2 \times 208 + 145 \]
\[ 208 = 1 \times 145 + 63 \]
compute using the rules of the syntax independent of the semantics

algorithms

GCD(561, 208)
Algorithm
A procedure to build 'everything' from a set of building blocks

We need to remember
• The building blocks
• The algorithms

12137 + 35823

What are the building blocks of addition?
The 'words' of addition

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
</tbody>
</table>
There is a trade-off between

the size of building blocks: memories

and the complexity of the algorithm: composition of the memories
Remembering a large memory with a composition of small memories

Syntax Boxes

BIG from small
### Syntax Boxes (s-box)

**Inputs**
- a
- b

**S-Box**

**Outputs**
- o

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
</tbody>
</table>
Binary Syntax Boxes (s-box)
Binary Syntax Boxes (s-box)

inputs

outputs

0

0

a
b

0
0
0
0
Binary Syntax Boxes (s-box)
Binary Syntax Boxes (s-box)
Suggest a name for this memory box?
Suggest a name for this memory box?

Can we remember $\max(a, b, c)$ with $\max(a, b)$?

\[ o = \max(a, b) \]
Can we remember \(\max(a,b,c)\) with \(\max(a,b)\)?

\[ o = \max(a,b) \]

**Composition:**
- build **big** s-boxes from **small** s-boxes.

\[ o = \max(a,b,c) \]
How to compute the following (XOR)?

Can we remember \( w \) with the max s-box?

\[
o = \max (a, b)
\]

\[
\begin{array}{ccc}
a & b & o \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

NO Why not?
Can we remember \( w \) with the max s-box?

Composition:
build big s-boxes from small s-boxes

The output of every small s-box is bigger or equal to its inputs

The output of the big s-box must be bigger or equal to its inputs

\[
\begin{array}{ccc}
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{ccc}
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

NO

Why not?
A Magic (Universal) Box

A binary s-box that can compute any binary s-box?
A Magic Box

Can you compute the following with the magic box?

Can you compute the \( \min(x, y) \)?
Hint 1: A Magic Box

Can you compute the following with the magic box?

\[
\min(x, y)
\]
Hint 2: A Magic Box

Can you compute the following with the magic box?

\[
\begin{array}{ccc}
a & b & m \\
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{ccc}
a & b & m \\
x & y & o \\
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
\end{array}
\]

\[
\min(x, y)
\]
A Magic Box

\[
\begin{array}{c}
\text{min}(x,y)
\end{array}
\]
A Magic Box

\[
\begin{array}{c}
\min(x, y)
\end{array}
\]
Can you compute the following with the m-box?

\[
\begin{array}{ccc}
a & b & m \\
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{ccc}
x & y & o \\
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

\[
M = \begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 1 & 1 \\
1 & 0 & 0 \\
\end{bmatrix}
\]

\[
\max(x, y)
\]
A Magic Box

\[ \max(x, y) \]
A Magic Box

max(x, y)
A Magic Box

\[ \text{max}(x, y) \]

\[ M \]

\[
\begin{array}{c}
\hline
| a | b | m |
\hline
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
\hline
\end{array}
\]

\[
\begin{array}{c}
\hline
| x | y | o |
\hline
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
\hline
\end{array}
\]
A Magic Box

\[ \max(x, y) \]
m-Box: A **two input** binary syntax box that can compute **any** (two input) binary syntax box? 

How many different binary 2-input s-boxes? $2^4 = 16$

How will you prove it?
Syntax Boxes
proof of universality
4 Useful Boxes

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>o</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>o</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The diagram shows a Boolean function M with inputs a, b, m, x, y, and output o. The function diagram includes a calculation of 1-y and the application of the Boolean function M. The input values for a, b, m, x, and y are presented in two matrices, with a highlighted section showing the values 0, 0, 1, 1, 1. The output values for o are also presented in a matrix, with highlighted sections showing 0, 0, 1, 0, 0.
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ y \]

\[ 1-y \]

\[ M \]

\[ o \]

\[ x \]

\[ y \]

\[ o \]

\[
\begin{array}{ccc}
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 1 & 1 \\
\end{array}
\]
4 Useful Boxes

- \( \min(x, y) \)
- \( 1 - y \)
- \( 1 \)
- \( \max(x, y) \)

So what?

Need to prove:

any \textbf{(two input)} binary syntax box can be computed by the 4 Useful Boxes
An Arbitrary Two Input Box

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>o</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>*</td>
</tr>
</tbody>
</table>

Two 1-input boxes!

- x = 0 then
- x = 1 then

What are the possible values of p?

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
An Arbitrary Two Input Box

\[
\begin{align*}
\min(x, y) & \quad 1-y & \quad 1 \\
\max(x, y) & \quad y
\end{align*}
\]

What are the possible values of \( y \)?

Can we compute it with the m-box?
A Selector Box

x = 0 then

x = 1 then

How can you compute this box?

selector

x

0

0 1 0 1

x y o

0 0 1 1
x = 0 then

x = 1 then

A Selector Box

\[
\begin{align*}
\text{min}(a, b) & \quad \text{1} - x \\
\text{min}(a, b) & \quad x
\end{align*}
\]

\[
\max(a, b) \quad 0
\]
x = 0 then

x = 1 then

\( \text{min}(a,b) \)

\( \text{max}(a,b) \)

\( \text{min}(a,b) \)
\(x = 0\) then

A Selector Box

\(x = 1\) then

\[\begin{align*}
\min(a, b) & \rightarrow 1 \\
\min(a, b) & \rightarrow 0 \\
\max(a, b) & \rightarrow 0
\end{align*}\]

QED
How to compute the following (XOR)?

\[
\begin{align*}
\text{x} &= 0 \text{ then } 1 \\
\text{x} &= 1 \text{ then } 0 \\
\end{align*}
\]
How to compute the following (XOR)?

- If \( x = 0 \) then \( 1 \)
- If \( x = 1 \) then \( \frac{1-y}{0} \)

Output:

\[
\begin{array}{c}
\text{selector} \\
0 \\
\end{array}
\]
\[ y = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x = 1 \end{cases} \]

\[ M = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \]

\[ \text{selector} \]

\[ 1-y \]

\[ x \]

\[ y \]

\[ 0 \]
Does the magic continue?

Given a 2-input binary box that can compute **any** 2-input binary box.

Can it compute **any** 3-input binary box?
3-input binary s-box

How many different binary 3-input s-boxes?

$2^8 = 256$
### 3-input binary s-box

**Two 2-input boxes!**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>o</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>*</td>
</tr>
</tbody>
</table>

- **z = 0 then**
- **z = 1 then**
All boxes can be computed with $M$
Does the magic continue?

Given a magical box for any 2-input binary box:

We proved that it is magical for any 3-input binary box!

Is it magical for any n-input binary box????

Proof by induction on the number of inputs

Yes!!!!
All boxes can be computed with $M$.

$(n-1)$ inputs

**selector**

$z = 0$ then

$z = 1$ then

0
Questions about algorithms and building blocks?

**Feasibility**

Given a set of building blocks: What can/cannot be constructed?

**Efficiency and complexity**

Size: If feasible, how many blocks are needed?

Time: How long will it take to complete the construction?
A word that is associated with the following?
A word that is associated with the following?

Face
A word that is associated with the following?

Face
Homework # 2

Due Thursday, April 23, 2015, at 2:30 PM
Collaboration and discussions are not allowed on Problem 1
and are allowed and encouraged on Problem 2

You have one week!
1. **Base-b Positional Systems** *(Collaboration is not allowed on this Problem)*

Here you will discover some interesting facts about positional number systems. Recall that an integer \( N \) represented in base \( b \) is denoted by

\[
N = (d_m, d_{m-1}, \ldots, d_1, d_0)_b
\]

with \( N = \sum_{i=0}^{m} d_i \cdot b^i \) and \( 0 \leq d_i \leq b - 1 \).

*When answering the questions, you need to show your work.*

(a) Given that \((292)_{10} = (1204)_{b} \), determine the value of \( b \).

(b) What are all the possible solutions for \( b \) and \( c \)?

   i. \((121)_{b} = (100)_{c}\)

   ii. \((50)_{b} = (43)_{c}\)

(c) What are all the possible solutions for \( b \)?

   i. \((1234)_{b} + (5432)_{b} = (6666)_{b}\)

   ii. \((41)_{b} \div (3)_{b} = (13)_{b}\)

   iii. \(\sqrt{(41)}_{b} = (5)_{b}\)
1. **Base-b Positional Systems** (*Collaboration is not allowed on this Problem*)

Here you will discover some interesting facts about positional number systems. Recall that an integer $N$ represented in base $b$ is denoted by

$$N = (d_m, d_{m-1}, \ldots, d_1, d_0)_b$$

with $N = \sum_{i=0}^{m} d_i \cdot b^i$ and $0 \leq d_i \leq b - 1$.

*When answering the questions, you need to show your work.*

(d) What is the base-60 representation of $(0; 10011)_2$? Recall that the symbol $;$ represents the fractional point, where the weights following the fractional point are $b^{-i}$ and $i \geq 1$. Show your work.
1. **Base-b Positional Systems** (*Collaboration is not allowed on this Problem*)

Here you will discover some interesting facts about positional number systems. Recall that an integer \( N \) represented in base \( b \) is denoted by

\[
N = (d_m, d_{m-1}, \ldots, d_1, d_0)_b
\]

with \( N = \sum_{i=0}^{m} d_i \cdot b^i \) and \( 0 \leq d_i \leq b - 1 \).

When answering the questions, you need to show your work.

(e) Now we explore the base-\( b \) representation of \( \pi \). The best 6 digits approximation of \( \pi \) in base-10 is \( \pi = (3; 14159)_{10} \).

Find the best approximation of \( \pi \) using a base-60 representation with 6 digits. What is the difference (in base-60) between these two representations of \( \pi \)?

\[
\pi
\]

Difference in approximation - 7 digits base 60
2. **Base-$b$ no-0 Positional Systems** *(Collaboration is allowed on this Problem)*

In class we studied the base-10 no-0 positional system where a positive integer $N$ is represented as follows:

$$N = \sum_{i=0}^{m} d_i 10^i$$

with $d_i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, A = 10\}$ for all $0 \leq i \leq m$.

Examples, $N = (10)_10 = (A)$, $N = (100)_10 = (9A)$ and $N = 0 = \text{ (the blank string) }$.

We expect you to solve the following problems with syntax-based procedures (as was demonstrated in class) and not by base-conversion procedures. Show your work and explain your methods.

(a) What is the base-10 no-0 representation of $(21300093081)_10$?

What is the base-10 representation of $(AA7A32A99A)$?
2. **Base-\(b\) no-0 Positional Systems** (*Collaboration is allowed on this Problem*)

In class we studied the base-10 no-0 positional system where a positive integer \(N\) is represented as follows:

\[
N = \sum_{i=0}^{m} d_i 10^i
\]

with \(d_i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, A = 10\}\) for all \(0 \leq i \leq m\).

Examples: \(N = (10)_{10} = (A)\), \(N = (100)_{10} = (9A)\) and \(N = 0 = \) (the blank string).

We expect you to solve the following problems with syntax-based procedures (as was demonstrated in class) and not by base-conversion procedures. Show your work and explain your methods.

(b) Suggest an analogous base-2 no-0 representation.

*Hint: the number \((4)_{10}\) is represented by \((100)_{2}\) in base-2 and by \((12)\) in the base-2 no-0 representation.*

Represent the base-2 number \((101101101000110)_{2}\) in base-2 no-0 representation.

Represent the base-2 no-0 number \((11221112122112)\) in the base-2 representation.
2. **Base-\( b \) no-0 Positional Systems** *(Collaboration is allowed on this Problem)*

In class we studied the base-10 no-0 positional system where a positive integer \( N \) is represented as follows:

\[
N = \sum_{i=0}^{m} d_i 10^i
\]

with \( d_i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, A = 10\} \) for all \( 0 \leq i \leq m \).

Examples, \( N = (10)_{10} = (A) \), \( N = (100)_{10} = (9A) \) and \( N = 0 = \) (the blank string).

We expect you to solve the following problems with syntax-based procedures (as was demonstrated in class) and not by base-conversion procedures. Show your work and explain your methods.

(c) Suggest an analogous base-60 no-0 representation.

Represent the number \((33, 0, 0, 17, 0, 49)_{60}\) in the base-60 no-0 representation. Note that the notion of no-0 relates to the digit base-60 that corresponds to 0.