Homework 2

September 12, 2025

1. (50 pts) We have the following code "Classic_Bit_Multiplication.c" which implements the old version of bit multiplication. Your task are three for this:

   (a) Implement the Gauss Trick for the code using the ideas.

   (b) As you can see we have something quite simple:

   ```
   if (n == 1){
     return a*b;
   }
   ```

   This uses the ALU for integer multiplication, but it can be substituted for something more efficient using xor because. For example, we can use xor for unsigned integer addition:

   ```
   unsigned int bitwise_add(unsigned int x, unsigned int y) {
       unsigned int carry;
       while (y != 0) {
           carry = x & y;
           x = x ^ y;
           y = carry << 1;
           }
           return x;
   }
   ```

   (c) Please compare the time complexity between both implementations by using the "sys/time.h" using gettimeofday() for higher precision.

   (d) Question why using **unsigned long long** when the **unsigned int** is of 32 bits?

2. (50 pts) In the case of the code Basic_Tree_Operations.c, you have the basic of a really skewed tree to the right. Please do the following:

   (a) Read the data from a text file

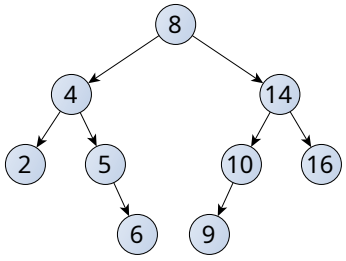
   (b) Create a procedure using a binary search to create a mostly balanced tree by extracting the middle of each segment as for example:

   i. | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 14 | 16 | $->$ 8 insert in the correct position of the binary tree.

   ii. | 2 | 4 | 5 | 6 | $->$ 4 or 5

   iii. | 9 | 10 | 14 | 16 | $->$ 10 or 14

   Thus, we have the following tree:



   (c) Free the memory at the heap because of using malloc for the binary tree. In my skewed version there is a partial idea of the this.