Practice Quiz 1

- Do not open this quiz booklet until you are directed to do so.
- This quiz starts at 2:35 P.M. and ends at 3:55 P.M. It contains 4 problems, some with multiple parts. The quiz contains 12 pages, including this one. You have 80 minutes to earn 100 points.
- This quiz is closed book. You may use one handwritten $8\frac{1}{2}'' \times 11''$ crib sheet. No calculators are permitted.
- When the quiz begins, write your name on every page of this quiz booklet in the space provided.
- Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem. Do not put part of the answer to one problem on the back of the sheet for another problem, since the pages will be separated for grading.
- Do not spend too much time on any problem. Read them all through first and attack them in the order that allows you to make the most progress.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.
- Good luck!

Problem 1. Integer Multiplication [15 points]

The following algorithm multiplies two nonnegative integers A and B:

```
MULT(A, B)
P ← 0
while A ≠ 0
    do if A mod 2 = 1
          then P ← P + B
       A ← ⌊A/2⌋
       B ← 2B
return P
```

Let $A^{(0)}$, $B^{(0)}$, and $P^{(0)}$ be the values of A, B, and P, respectively, immediately before the loop executes, and for $k \geq 1$, let $A^{(k)}$, $B^{(k)}$, and $P^{(k)}$ be the values of these variables immediately after the kth iteration of the loop. Give a loop invariant that can be used to prove the correctness of the algorithm. *You need **not** actually prove correctness.*

Problem 2. True or False, and Justify [50 points] (10 parts)

Circle **T** or **F** for each of the following statements to indicate whether the statement is true or false, respectively. If the statement is correct, briefly state why. If the statement is wrong, explain why. The more content you provide in your justification, the higher your grade, but be brief. Your justification is worth more points than your true-or-false designation.

T F The solution to the recurrence

$$T(n) = 100\,T(n/99) + \lg(n!)$$

is $T(n) = \Theta(n \lg n)$.

T F Radix sort works correctly even if insertion sort is used as its auxiliary sort instead of counting sort.

T F If bucket sort is implemented by using heapsort to sort the individual buckets, instead of by using insertion sort as in the normal algorithm, then the worst-case running time of bucket sort is reduced to $\Theta(n \lg n)$.

T F An adversary can present an input of n distinct numbers to RANDOMIZED-SELECT that will force it to run in $\Omega(n^2)$ time.

T F The information-theoretic (decision-tree) lower bound on comparison sorting can be used to prove that the number of comparisons needed to build a heap of n elements is $\Omega(n \lg n)$ in the worst case.

T F Sorting 6 elements with a comparison sort requires at least 10 comparisons in the worst case.

T F The sum of the smallest $\sqrt{n}$ elements in an unsorted array of n distinct numbers can be found in $O(n)$ time.

T F The collection $\mathcal{H} = \{h_1, h_2, h_3\}$ of hash functions is universal, where the three hash functions map the universe $\{A, B, C, D\}$ of keys into the range $\{0, 1, 2\}$ according to the following table:

x	$h_1(x)$	$h_2(x)$	$h_3(x)$
A	1	0	2
B	0	1	2
C	0	0	0
D	1	1	0

T F Let X be an indicator random variable such that $\Pr\{X = 1\} = 1/2$. Then, we have $\mathrm{E}[X(1 - X)] = 1/4$.

T F Suppose that a 3-input sorting network correctly sorts the sequences $\langle 2, 3, 8 \rangle$, $\langle 3, 8, 2 \rangle$, and $\langle 8, 2, 3 \rangle$. Then, it also correctly sorts all sequences of 3 numbers. (*Hint: Apply threshold functions to the sequence elements.*)

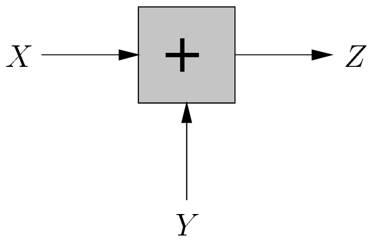
Problem 3. Pop Count [35 points] (5 parts)

Some computers provide a ***population-count***, or ***pop-count***, instruction POPC that determines the total number of bits in a word that are "1." Specifically, if an n-bit word is $a = \langle a_0 a_1 \cdots a_{n-1} \rangle$, then

$$\text{PopC}(a) = \sum_{i=0}^{n-1} a_i \ .$$

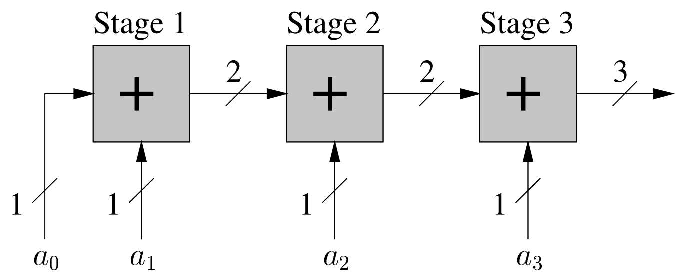
For example, $\text{PopC}(\langle 01000101 \rangle) = 3$.

This problem explores how to implement the POPC instruction as a circuit. The basic building block we shall use is an ADDER component that takes in two binary values X and Y and produces their sum $Z = X + Y$:



Since wiring is an important contributor to the expense of an implementation, we shall attempt to minimize the number of wires required to connect the inputs, components, and output of a pop-count circuit. A cable of w wires can convey values in the range from 0 to $2^w - 1$, inclusive.

Professor Blaise has invented a pop-count circuit, composed of $n - 1$ ADDER's, to implement POPC on an n-bit word $\langle a_0 a_1 \cdots a_{n-1} \rangle$. Here is the professor's circuit for $n = 4$:



Each stage i of this circuit adds the bit a_i into a running sum, which it forwards to stage $i+1$. Each cable of wires in the figure is labeled with the number of its constituent wires.

(a) Argue that the stage-i ADDER requires $\Theta(\lg i)$ output wires.

(b) For an input word with $n = 4$ bits, the total number of wires in the professor's pop-count circuit is $W(n) = 11$ wires. Explain briefly why the recurrence

$$W(n) = W(n-1) + \Theta(\lg n)$$

accurately describes the total number of wires in an n-input circuit. Give a good asymptotic lower (big-Ω) bound for $W(n)$, and briefly justify your answer.

A divide-and-conquer pop-count circuit operating on an n-bit word $\langle a_0 a_1 \cdots a_{n-1} \rangle$ can be constructed by using an ADDER component to combine the recursively computed pop-count of the first $n/2$ bits with the recursively computed pop-count of the last $n/2$ bits.

(c) Draw a picture of such a divide-and-conquer pop-count circuit on 4 inputs. Label the number of wires comprising each cable, as was done for Professor Blaise's circuit. How many wires does the divide-and-conquer circuit require for $n = 4$?

(d) Give a recurrence that describes the total number $W(n)$ of wires required by the divide-and-conquer pop-count circuit for an n-bit word. Give a good asymptotic upper (big-O) bound on $W(n)$, and briefly justify your answer.

(e) Does Professor Blaise work at MIT or Harvard? Why?