ECE G205 Fundamentals of Computer Engineering
Fall 2004

Homework 1: Due in class Wednesday September 15 2004

- This test contains 10 problems. They allow you to earn 100 points.
- Show your work, as partial credit can 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**.
- **No late submissions will be accepted.**
- Only homework returned in a 9in $\times$ 12in envelope will be accepted. (If you cannot find such envelope, ask the Instructor.) Please, write your name and the class name (ECE G205) on the envelope (write clearly, please).

Write your name here: ____

1. *Concept: Mathematical Induction and Summations*

   (a) Prove the following:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

   (b) What is wrong with the following theorem and proof?

   Theorem: All babies have the same color eyes.

   Proof. The base case of induction, when $n = 1$, is obvious. Suppose now that every set of n babies have the same color eyes (inductive step). Consider a set of $n+1$ babies.

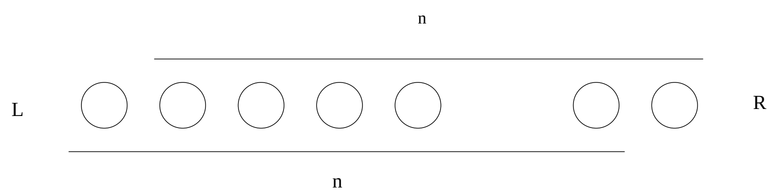

Figure 1: Two sets of n babies.

   We may assume by induction that the set L of n babies (the n leftmost babies in Figure 1) has babies with the same color eyes. Similarly all the n babies in the set R (the n rightmost babies) have the same color eyes. But then, evidently, all the $n+1$ babies in the figure have the same color eyes, for the leftmost and the rightmost babies have the same color eyes as the babies in between. Therefore, by mathematical induction, for every natural number n, for every set of n babies, all have the same color eyes. Since the set of all babies is such a set, the theorem is proved.

2. *Concept: Basic Set Theory*

   (a) Given two finite sets A and B define their union $A \cup B$ and their intersection $A \cap B$.

   (b) Show that for every finite set S with n elements, the set 2^S (i.e., the set of all possible subsets of S) has 2^n elements (i.e., that there are 2^n distinct subsets of S).

   (c) Show that there are as many odd numbers as natural numbers (i.e., show that the set of odd number is *countable*).

3. *Concept: Asymptotic Notation*

(a) Given a function $f : \mathbf{N} \to \mathbf{N}$, what is the meaning of $O(f(n))$?

(b) *True or False*. Answer the following questions.

- $n+3 \in \Omega(n)$
- $n+3 \in O(n^2)$.
- $n+3 \in \Theta(n^2)$.
- $2^{n+1} \in O(n+1)$.
- $2^{n+1} \in \Theta(2^n)$.

4. *Concept: Basic Graph Theory*

(a) Consider the definition of *undirected graph*, i.e., the couple $G = (V, E)$ where V is the a set of *vertices* and $E \subseteq V \times V$. Based on this definition define what is a *tree*.

(b) What is the maximum possible length of a *path* in a tree?

5. *Concept: Basic Data Structures*

(a) Define and compare the data structures *array* and *linked lists*.

(b) Describe how given two linked lists $\ell_1 = \langle a_1, a_2, \ldots, a_n \rangle$ and $\ell_2 = \langle b_1, b_2, \ldots, b_m \rangle$ these can be concatenated to form a single list $\ell_{1,2} = \langle a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m \rangle$.

6. *Concept: Sorting and Searching*

   (a) Describe and implement a sorting algorithm for arrays of n integers.

   (b) Describe and implement the searching for the *minimum* element in an array of n integers. Assume the array is *not* sorted. What changes if the array is sorted?

7. *Concept: Basic C commands and Their Computational Cost (with a hint of discrete math to boot)*

(a) What is the value returned by the following function? Express your answer as a function of *n*.

```
int pesky( int n ) {
  int r = 0;
  for( int i = 1; i < n; i++ )
    for( int j = 1; j < i + 1; j++ )
      for( int k = j; k < i + j + 1; k++ )
        r++;
  return r;
}
```

(b) Using the "big O" notation give the worst-case running time of function `pesky`.

8. *Concept: Functions and Parameter Passing in C*

   (a) What is a C function *prototype*, and why is it needed?

   (b) Describe all possible ways of passing parameters to a C function, and illustrate their differences.

9. *Concept: Scope of identifiers in C*

   (a) What is the scope of an identifier in C?

   (b) For the following program, state the scope (either function scope, file scope, block scope, or function prototype scope) of each of the following elements.

      i. The variable *x* in main.
      ii. The variable *y* in cube.
      iii. The function cube.
      iv. The function main.
      v. The function prototype for cube.
      vi. The identifier *y* in the function prototype for cube.

```
#include <stdio.h>
int cube( int y );
int main() {
  int x;
  for( x = 1; x <= 10;  x++ )
    printf( "%d\n", cube( x ) );
  return 0;
}
int cube( int y ) {
  return y * y * y;
}
```

10. *Concept: Logical Expressions in C*

(a) In the following C logical expression:

```
( a == b ) || ( a + b > 32 )
```

which of the two Boolean conditions is evaluated first? If the first condition evaluates to true, is the second condition evaluated at all?

(b) Based on the previous answer what would be a good performance tip for those who write C code?

(c) What is the value of this expression if *a* equals 1 and *b* equals 2?