Discussion #1 Outline

1. Introductions
2. Announcements
   a. Your Office Hours
      i. Times
      ii. Place – MU3NE
   b. Website
      i. [http://www.eecs.umich.edu/courses/f04/index.html](http://www.eecs.umich.edu/courses/f04/index.html)
   c. Homework/Project assignment and due dates
      i. Homework 1 – Assigned Tues. Sept 14, Due Tues. Sept 28 in dropbox – need cover sheet from website
3. Pseudocode
   a. What is pseudocode? Why do we need it?
   b. Specifics of Pseudocode
   c. C++ to Pseudocode example (as class)
   d. Pseudocode to C++ example (in small groups, 2-3)
   e. Opportunity for Questions and Clarifications
4. Big-O Fallacies
   a. 6 big-O fallacies
   b. Opportunity for Questions and Clarifications
5. Open Question Time
   a. Take questions about lectures, homework assignment, project assignment, etc.
Purpose of Pseudocode:
- Writing for human, not computer
- High-level description of an algorithm
- More structured than prose, less detailed than C++
- Preferred notation for describing algorithms
- Hides program design issues

Pseudocode Details:
- Method declaration
  Algorithm method (arg [, arg...])
  Input ...
  Output ...
- Control flow
  if ... then ... [else ...]
  while ... do ...
  repeat ... until ...
  for ... do ...
  Indentation replaces braces
- Array indexing
  A[i] is ith cell in array A
  A is from A[0] to A[n-1]
- Method/Function call
  method (arg [, arg...])
- Return value
  return expression
- Expressions
  ← Assignment
  (like = in C++)
  ← Equality testing
  (like == in C++)
  $n^2$ Superscripts and other math formatting allowed
Example 1: C++ to Pseudocode

```c
int arrayMax(int A[ ], int n)
{
    int currentMax = A[0];
    for (int i = 1; i < n; i++)
        if (currentMax < A[i])
            currentMax = A[i];
    return currentMax;
}
```

Algorithm arrayMax(A, n)
Input: array $A$ of $n$ ints
Output: max element of $A$

$\text{currentMax} \leftarrow A[0]$
for $i \leftarrow 1$ to $n - 1$ do
    if $A[i] > \text{currentMax}$ then
        $\text{currentMax} \leftarrow A[i]$
return $\text{currentMax}$
Example 2: Pseudocode to C++

In groups of 2 or 3, turn this Pseudocode into C++ code that compiles…

Algorithm arrayFind(x, A, n)
   Input: An element x and an n-element array, A.
   Output: The index I such that x=A[i] or -1 if no element of A is equal to x.
   for i ← 0 to n-1 do
      if x=A[i] then
         return i
   return -1

Answer:

int arrayFind(int x, int A[], n) {
   for(int i=0; i<n; i++) {
      if(x==A[i])
         return i;
   }
   return -1;
}
Big-O Fallacies

**Fallacy 0:**
If \( f(n) = O(g(n)) \) \( \Rightarrow f(n) = g(n) \)

**Fallacy 1:**
Let \( f_1(n) = h(n^2) \) and \( f_2(n) = h(n^2) \) \( \Rightarrow f_1(n) = f_2(n) \)
Therefore
\( f_1(n) = O(n^2) \) and \( f_2(n) = O(n^2) \) \( \Rightarrow f_1(n) = f_2(n) \)

**Fallacy 2:**
\( f(n) = O(g(n)) \) \( \Rightarrow g(n) = O^{-1}(f(n)) \)
(There is no such thing as \( O^{-1}() \) function.)

**Fallacy 3:**
Let \( f(n) = g_1(n) \times g_2(n) \).
If \( f(n) \leq C \times g_1(n) \) for all integers \( n \geq 0 \) and \( C = g_2(n) \), then \( f(n) = O(g_1(n)) \)

**Fallacy 4:**
Let \( f_1(n) = O(g_1(n)) \) and \( f_2(n) = O(g_2(n)) \) for non-negative functions and \( n \geq 0 \).
If \( g_1(n) < g_2(n) \), then \( f_1(n) < f_2(n) \).

**Fallacy 5:**
Let \( f_1(n) = O(g_1(n)) \) and \( f_2(n) = O(g_2(n)) \) for non-negative functions and \( n \geq 0 \).
If \( g_1(n) < g_2(n) \), is there an integer \( n_0 > 0 \) for which \( f_1(n_0) < f_2(n_0) \).