CSE 373 -- Autumn 1999 -- Assignment #1

(*due Friday, October 8*)

1. **Prove that the following is true when *n* is odd: $1 + 2 + 3 + ... + n = (n*(n+1))/2$**
   Your proof need not be particularly formal, but it must be correct and clear (*i.e.*, there should be no doubt in the grader's mind that you understand why the relationship is true for odd values of *n*).

2. **Consider the following implementation of `fib()` which takes a slightly different approach than those we discussed in Wednesday's lecture:**

```
static int* memo;

static int fib_help(int x) {
  if (memo[x] != -1) {
    return memo[x];
  } else if (x <= 1) {
    return 1;
  } else {
    return fib_help(x-1) + fib_help(x-2);
  }
}


int fib(int x) {
  int i;
  int retval;

  memo = new int[x+1];
  for (i=0; i<=x; i++) {
    memo[i] = -1;
  }

  retval = fib_help(x);

  delete memo;

  return retval;
}
```

   (*a*) Asymptotically speaking, what is the running time of this implementation of `fib()`? Why? (Be sure to indicate what the problem size is characterized by)

   (*b*) How much space is required by the function call stack for this implementation of `fib()`, asymptotically? Why?

   (*c*) How much *total* space is required by this implementation of `fib()`, asymptotically? Why?

   (*d*) We discussed three potential drawbacks of recursion in lecture: the time required to make recursive calls that would not be needed in an iterative solution, the space required for the function call stack, and the possibility of performing redundant computation. Does this implementation of `fib()` have these drawbacks? Why or why not?

3. **Most children reach a stage where they like playing number-guessing games in which their parent thinks of a number between 1 and 10 (or perhaps an even bigger number) and they have to guess what the number is. After each guess, the parent tells them whether they are too low, too high, or whether they guessed it correctly.**

   At some point, many kids reach a different stage where they like to turn the game around and make their parents guess *their* number. This inevitably starts to drive parents crazy. Imagine a function that could save the parents of the world: `GuessNum()`. This function would take the parents' place in this situation, providing hours of entertainment for their child by guessing the number that s/he is thinking of until it figures it out (at which point it should print an appropriately victorious message). `GuessNum()` would have the prototype: `void GuessNum(int lo, int hi);` The parameters `lo` and `hi` indicate the range of values in which the number lies. For example, the initial call might be `GuessNum(1,10)`.

   (*a*) Write a recursive implementation of `GuessNum()` whose worst-case running time of $O(n)$.

   (*b*) Write a recursive implementation of `GuessNum()` whose worst-case running time of $O(\log n)$.

   For both routines, assume that we supply you with a routine `int CheckGuess(int guess);` that takes a guess, prints it for the user, asks the user whether it's too low, too high, or correct, and returns an integer indicating the user's response (-1 = too low, 1 = too high, 0 = correct). You may assume that the user won't cheat by changing their number during the game (that's an even later stage of child development...).

   Note that you don't have to code this on-line unless you want to -- writing it with pencil and paper is fine.

4. **Rank the following functions asymptotically. If multiple functions are asymptotically equivalent, indicate them as such:**

   $3n^3 + 2n^2 + n$
   $5*2^n$
   $2\log_5 n$
   $23\log n + 0.25n$
   25
   $1 + 2 + 4 + ... + n/2 + n$
   $5n^2$
   $2^n + 1000n^{10}$
   $2n^5$
   $n^2\log^2 n + 32n$
   $\sin(45)$
   $1 + 2 + 3 + ... + n$
   $128n + 5n\log_6 n$
   $\log_n n$
   $256n^3 + 1$
   $69254^{0.5}$
   $12n^2\log n$
   $5\log_2 n$

5. **Give the asymptotic running time of each of the following code fragments and justify your answer:**

(a)

```
{
  int i,j;

  for (i=0; i<n; i++) {
    for (j=0; j<3*n; j++) {
      cout << i << "," << j << endl;
    }
  }
}
```

(b)

```
{
  int i,j;

  for (i=0; i<n; i++) {
    for (j=i; j<n; j++) {
      cout << i << "," << j << endl;
    }
  }
}
```

(c)

```
{
  int i,j;

  for (i=0; i<n; i++) {
    for (j=i-1; j<=i+1; j++) {
      cout << i << "," << j << endl;
    }
  }
}
```

(d)

```
{
  int i,j;

  for (i=0; i<n; i++) {
    for (j=1; j<=exp(2,i+1); j++) {     // exp(x,y) = x^y
      cout << i << "," << j << endl;
    }
  }
}
```

6. **Exercise 2.2 (a)** (Say whether it is true or false and justify your answer)

7. **Exercise 2.10 (a)-(b)** (Justify your answer)