WORKSHOP 18: §8.6, 8.9, 8.10   MARCH 23, 2017

Name: ____________________

(1) Complete the following sketch of a proof by strong induction.
Theorem. *Define a sequence by $a_0 = 0$, $a_1 = 2$, and $a_{n+1} = 4(a_n - a_{n-1})$ for $n \geq 1$. Then a closed form for $\{a_n\}$ is $a_n = n \cdot 2^n$.*
Proof. Proof by strong induction on n.
Base case(s). ??????????????????? *(In particular, how many base cases are needed? This ties in to the lower bound on k you require for you inductive step, below.)*
Inductive step. Assume that k is an integer, $k \geq$???, and that for every $j \in \{0, \ldots, k\}$, we have $a_j =$?????. *[We want to show that $a_{k+1} = (k+1)2^{k+1}$.]*
We have

a_{k+1}	$=$?????	by the recurrence relation
	$=$?????	the inductive hypothesis *the lower bound you gave on k above is important here! Why?*
	$=$???	algebra
	$=$???	more algebra
	$=$???	more algebra if needed?
	$= (k+1)2^{k+1}$	algebra

which is what we wanted to show.
So, by PSMI *[the Principle of Strong Mathematical Induction]*, the theorem holds. □

(2) The sorting algorithm `mergesort` below is a well-known recursive algorithm.
Prove (using induction) that the time complexity of `mergesort` is $\Theta(n \log n)$. [1]

```
Input:   (a_1, ..., a_n) a sequence of real numbers
         n, the length of the sequence, n ≥ 1
Output:  (d_1, ..., d_n), the input sequence sorted in nondecreasing order.

If n = 1
   Return( (a_1) ) //a list of length 1 is already sorted

left := (a_1,...,a_⌊n/2⌋)
right := (a_⌊n/2⌋+1,...,a_n)
(b_1,...,b_⌊n/2⌋) := mergesort(left)
(c_1,...,c_⌈n/2⌉) := mergesort(right)
i := 1
j := 1
For k = 1 to n:
   If b_i ≤ c_j
      d_k := b_i
      i := i+1
   Else  //b_i > c_j
      d_k := c_j
      j := j+1
   End-if
End-for
Return (d_1, ..., d_n)
```

(3) Define a sequence by $a_1 = 1$, $a_n = 2a_{\lfloor n/2 \rfloor}$ for $n \geq 2$.

  (a) Compute the first several values of the sequence. Conjecture a closed form. (It should involve $\lfloor \log_2 n \rfloor$...)

  (b) Prove that your closed form is correct using strong induction. (You will need a result from Workshop 10: For any $x \in \mathbb{R}$ and $m \in \mathbb{Z}$, $\lfloor x + m \rfloor = \lfloor x \rfloor + m$.)

(4) Let $\{f_n\}$ denote the Fibonacci sequence. Prove that

$$f_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right]$$

for all n. Part of the reason this is true – and this will also save you some algebra in your proof – is that $\varphi = \frac{1+\sqrt{5}}{2}$ and $\psi = \frac{1-\sqrt{5}}{2}$ are both solutions to the equation $x^2 = x + 1$. *(No, it's not the prettiest formula. But you can do it!)*

Notes

[1] Technically, this code doesn't work; when we get to the end of one of the lists (b_i) or (c_j), then we keep trying to compare. That is, we test `If` $b_i \leq c_j$, but at some point in the `For` loop we have $i > \lfloor n/2 \rfloor$ or $j > \lceil n/2 \rceil$ and so either b_i or c_j isn't defined. To fix the problem would take a little code that would not significantly affect the time complexity, but would muddy the code. This should not affect your answer. (And if you want to be a stickler, I could just define $a_{\lfloor n/2 \rfloor + 1} = b_{\lceil n/2 \rceil + 1} = \infty$, in which case the code will work as desired.)