Problem 1 [40 points (10 points each)] Branching Patterns
Consider the following figure where an denotes the number of matches in a growing pattern and n ≥ 1.

1. What are the first five values of the sequence?
   a1 = 3 = 3 * (2^1 – 1)
   a2 = 9 = 3 * (2^2 – 1)
   a3 = 21 = 3 * (2^3 – 1)
   a4 = 45 = 3 * (2^4 – 1)
   a5 = 93 = 3 * (2^5 – 1)

2. Give the recursive definition for the value of an, that is, a formula for an in terms of a_{n-1}. Don't forget to also state the base case!
   Base case, a1 = 3 = 3 * 1
   $a_n – a_{n-1} = 3 * (2^n – 1) – 3 * (2^{n-1} – 1) = 3 * (2 * 2^n – 1 – 2^n + 1) = 3 * 2^{n-1}$
   $a_n = a_{n-1} + 3 * 2^{n-1}$
3. Find a closed-form formula for a_n.
   $a_n = 3 * (2^n – 1)$ for all n >= 1
4. Using your recursive definition from sub-problem (2), prove that your closed-form formula is correct (for all n ≥ 1) using a proof by induction.

Theorem:

For n ≥ 1, $a_n = 3 * (2^n – 1)$

Proof.
By induction on n.
Base case: n = 1.
Therefore, for n = 1, $a1= 3 * (2^1 – 1) = 3$
Inductive step: We will show that for any integer k that k >= 1, if $a_k= 3 * (2^k – 1)$, $a_{k+1} = 3 * (2^{k+1} – 1)$
While k >= 1:
$a_{k+1} = a_k + 3 * 2^k$
$= 3 * (2^k – 1) + 3 * 2^k$ by the inductive hypothesis obtaining from sub-problem (2)
$= 3 * 2^k + 3 * 2^k - 3$
$= 3 *(2* 2^k * -1)$
$= 3 * (2^{k+1} – 1)$
Therefore, $a_{k+1} = 3 * (2^{k+1} – 1)$

Problem 2 [40 points]: Sorting
An array contains the following sequence: 8, 14, 99, 0, 1, 7, 19, 4. We wish to sort the array in DESCENDING order from LARGEST to SMALLEST value. For both (a) INSERTION sort and (b) SELECTION sort, indicate for each iteration:
1. The state of the array (after a number is moved into its correct position)
2. The number of comparisons performed for that iteration
3. The number of swaps performed for that iteration

Let N be the size of the sequence, which is 8

For (a) DESCENDING INSERTION sort: Start with 8, 14, 99, 0, 1, 7, 19, 4

Iteration 1:
(compare: 1) 8 < 14
(swap: 1) 14, 8
List = 14, 8, 99, 0, 1, 7, 19, 4

Iteration 2:
(compare: 1) 8 < 99
(swap: 1) 8, 99
(compare: 2) 14 < 99
(swap: 2) 99, 14
List = 99, 14, 8, 0, 1, 7, 19, 4

Iteration 3:
(compare: 1) 8 > 0
List = 99, 14, 8, 0, 1, 7, 19, 4

Iteration 4:
(compare: 1) 0 < 1
(swap: 1) 1, 0
(compare: 2) 8 > 1
List = 99, 14, 8, 1, 0, 7, 19, 4

Iteration 5:
(compare: 1) 0 < 7
(swap: 1) 7, 0
(compare: 2) 1 < 7
(swap: 2) 7, 1
(compare: 3) 8 > 7
List = 99, 14, 8, 7, 1, 0, 19, 4

Iteration 6:
(compare: 1) 0 < 19
(swap: 1) 19, 0
(compare: 2) 1 < 19
(swap: 2) 19, 1
(compare: 3) 7 < 19
(swap: 3) 19, 7
(compare: 4) 8 < 19
(swap: 4) 19, 8
(compare: 5) 14 < 19
(swap: 5) 19, 14
(compare: 6) 99 > 19
List = 99, 19, 14, 8, 7, 1, 0, 4

Iteration 7:
(compare: 1) 0 < 4

(swap: 1) 4, 0
(compare: 2) 1 < 4
(swap: 2) 4, 1
(compare: 3) 7 > 4
List = 99, 19, 14, 8, 7, 4, 1, 0

DESCENDING INSERTION sort:7 iterations, 18 comparisons, 13 swaps.

For (a) DESCENDING SELECTION sort: Start with 8, 14, 99, 0, 1, 7, 19, 4

Iteration 1:
(compare: 1) 8 < 99
(swap: 1) 99, 8
List = 99, 14, 8, 0, 1, 7, 19, 4

Iteration 2:
(compare: 1) 14 < 19
(swap: 1) 19, 14
List = 99, 19, 8, 0, 1, 7, 14, 4

Iteration 3:
(compare: 1) 8 < 14
(swap: 1) 14, 8
List = 99, 19, 14, 0, 1, 7, 8, 4

Iteration 4:
(compare: 1) 0 < 8
(swap: 1) 8, 0
List = 99, 19, 14, 8, 1, 7, 0, 4

Iteration 5:
(compare: 1) 1 < 7
(swap: 1) 7, 1
List = 99, 19, 14, 8, 7, 1, 0, 4

Iteration 6:
(compare: 1) 1 < 4
(swap: 1) 4, 1
List = 99, 19, 14, 8, 7, 4, 0, 1

Iteration 7:
(compare: 1) 0 < 1
(swap: 1) 1, 0
List = 99, 19, 14, 8, 7, 4, 1, 0

DESCENDING SELECTION sort: 7 iterations, 7 comparisons, 7 swaps.

Problem 3 [20 pts]: MergeSort

Show me that you understand how MergeSort works by filling in all the boxes and drawing all the remaining arrows in the figure below. Again, we are sorting in DESCENDING order using the same starting sequence from problem 3.