Name: ____

1. (1 pt.)
   - **Read all material carefully.**
   - You may refer to your books, papers, and notes during this test.
   - E-books may be used subject to the restrictions noted in class.
   - No computer or network access of any kind is allowed (or needed).
   - Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
   - Use class and textbook conventions for notation, algorithmic options, etc.
   - There is an one extra-credit question (marked with ⋆). It is harder than the rest.
   - Write your name in the space provided above.

2. (9 pts.) Fill in the blank entries in the following table (**extending it** as needed), indicating the number of runs on each of the five tapes used in a *polyphase merge-sort of order 5*. Row n of each table summarizes the distribution of runs on the tapes immediately following the nth merge, with the 0th row summarizing the initial distribution of runs (before any merges).

	# runs on tape					
merge	1	2	3	4	5	6
0	20	6	13	9	4	0
1						
2						
3						

. . .

3. (15 pts.) Use *merge-based insertions* to insert the keys

   $15, 7, 10, 3, 2, 8, 9, 5, 4, 1$

   into an initially empty *skew heap*. Then perform three merge-based *deleteMin* operations. Depict the state of the tree *after each operation*.

[additional space for answering the earlier question]

$15, 7, 10, 3, 2, 8, 9, 5, 4, 1$

4. (15 pts.) Repeat all parts of Question 3 using a *pairing heap* instead of a skew heap. Reminder: Use precisely the textbook's method, and depict the left-to-right and right-to-left phases clearly.

   $15, 7, 10, 3, 2, 8, 9, 5, 4, 1$

[additional space for answering the earlier question]

$15, 7, 10, 3, 2, 8, 9, 5, 4, 1$

5. (5 pts.) Fill in the following table based on **the textbook's definition and notation** for B-trees,[1] with parameters $M = 4$ and $L = 3$. [Hint: Check the use of M and L carefully.]

node type:	leaf	non-leaf root	non-leaf non-root
min. number of keys:			
max. number of keys:			
min. number of children:			
max. number of children:			

6. (5 pts.) **Using the textbook's definition and notation** for B-trees, depict **all** B-trees with parameters $M = 4$ and $L = 3$ that contain exactly five records, with keys: $1, 2, 3, 4, 5$. Assume that keys within each B-tree node are always stored in sorted order. Explain briefly why there are no other trees satisfying the requirements.

[1] Mark Allen Weiss, *Data Structures and Problem Solving Using Java*, 4th edition (Addison-Wesley, 2010), §19.8, p. 756.

7. (10 pts.) Repeat Question 6 for the eight records, with keys $1, 2, \ldots, 8$. Explain briefly why there are no other trees satisfying the requirements.

[additional space for answering the earlier question]

8. (10 pts.) ⋆ A *comparison-sorting* method is a sorting method that does not use any properties of the input data other than the fact that a pair of items can be compared to determine which one is smaller. Provide methods to comparison-sort n items using the fewest comparisons, for each $n = 5, 6, 7$. Explain your methods clearly, and prove that no method can sort using fewer comparisons. Note that this question concerns the **precise number** of comparisons (e.g., 10, 17) not asymptotics (e.g., $O(n \log n)$, $O(n^2)$).