Name: ____

1. (1 pt.)
   - **Read all material carefully.**
   - *If in doubt whether something is allowed, ask, don't assume.*
   - 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.

   Write your name in the space provided above.

2. (14 pts.) Trace the execution of the FIND-MAX-CROSSING-SUBARRAY algorithm on the array `A` depicted below, with the arguments `low`, `mid`, and `high` equal to 1, 5, and 10, respectively.

`i`:	1	2	3	4	5	6	7	8	9	10
`A[i]`:	88	19	9	-66	-2	116	-56	-12	87	101

   List the values of *sum* and *left-sum* after each iteration of the first for-loop of the algorithm. Similarly, list the values of *sum* and *right-sum* after each iteration of the second for-loop.

3. (15 pts.) Depict the recursion tree that outlines the recursive calls made by the FIND-MAXIMUM-SUBARRAY algorithm when invoked on the array of Question 2 (repeated below), with `low` and `high` equal to 1 and 10, respectively. The nodes of the tree should be labeled with the function invoked (FIND-MAXIMUM-SUBARRAY or FIND-MAX-CROSSING-SUBARRAY and the edges should connect each function's node to the node of its invoker.

`i`:	1	2	3	4	5	6	7	8	9	10
`A[i]`:	88	19	9	-66	-2	116	-56	-12	87	101

4. (10 pts.) List all *derangements* of the sequence $1, 2, 3, 4$.

5. (10 pts.) Let $!n$ denote the number of derangements of a sequence of n distinct items. Prove or disprove: $!n = (n-1)(!(n-1)+!(n-2))$ for $n > 1$.

6. (10 pts.) Prove or disprove: The following algorithm generates a *uniform random permutation* of an array `v` when invoked as `foo(v)`. (The function `Random(a,b)` is as defined in the textbook.)

```
foo(v) {
  n = v.length
  bar(v, n, 1, n)
}

bar(v, n, lo, hi) {
  if lo < hi then {
     if Random(0,1) < 1 then {
       swap v[lo] with v[Random(1,n)]
       foo(v, lo + 1, hi)
     }
     else {
       swap v[hi] with v[Random(1,n)]
       foo(v, lo, hi - 1)
     }
  }
}
```

[additional space for answering the earlier question]