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.
   - There is an extra-credit question (marked with $\star$). It is harder than the rest.
   - Budget your time: roughly one minute per point.

   Write your name in the space provided above.

2. (9 pts.) For the following mapping of rod lengths to prices, how many recursive invocations of CUT-ROD does the recursive top-down cut-rod algorithm make? *Provide an exact numerical answer along with an explanation.* [Hint: You do not need to solve the cut-rod instance.]

length:	1	2	3	4	5	6	7	8	9	10	11	12
price:	4	7	9	14	18	22	30	30	28	38	40	44

3. (10 pts.) Solve the following recurrences. *Clearly state the methods you use for your solutions and outline their key steps.* (Show your work.)

   (a) $T(n) = 2T(n/2) + 3n + 1$

   (b) $S(n) = 7S(n/2) + 8n\sqrt{n}$

[additional space for answering the earlier question]

4. (10 pts.) Trace the operation of the LCS-Length algorithm on the following sequences.

```
A C B A A B A
C B A C A A B
```

Depict the state of the b and c arrays (1) after four iterations of the outer nested loop and (2) at the end of the algorithm.

[additional space for answering the earlier question]

5. (10 pts.) Trace the operation of the PRINT-LCS algorithm on the result of Question 4. Provide the arguments for each of recursive call of PRINT-LCS.

6. (10 pts.) Depict tables similar to those in Figure 15.5 of the textbook for MATRIX-CHAIN-ORDER on the following input:

matrix:	A_1	A_2	A_3	A_4	A_5
dimension:	10×30	30×50	50×100	100×40	40×30

7. (10 pts.) Depict the *first three levels* of the recursion tree that outlines the recursive calls made by the FIND-MAXIMUM-SUBARRAY algorithm when invoked on the following array, 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 (M) or FIND-MAX-CROSSING-SUBARRAY (X).

   The *edges* should connect each function's node (child) to the node of its invoker (parent).

`i`:	1	2	3	4	5	6	7	8	9	10
`A[i]`:	88	-1	-11	-23	43	-6	8	-19	-58	50

[additional space for answering the earlier question]