

# SIXTH QUIZ

You have 15 minutes from the start of class to complete this quiz. Give partial answers if you can't give complete ones. Read the questions with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

## Problem 1 (8 points)

Here is a recursive procedure:

```
void asterisk (int n)
{
    System.out.println( "*" );
    if (n > 0)
    {
        Asterisk (n-1);
        Asterisk (n-1);
    }
}
```

(a) (2 points) How many asterisks will `asterisk(2)` print?

(b) (4 points) Give the recurrence relation that describes the amount of time this function takes to finish, given that its parameter is an integer  $n \geq 0$ . Assume that each `println` takes one unit of time and the boolean expression `n>0` takes so little time we can ignore it.

$$A_0 =$$

$$A_n =$$

(c) (2 points) What is the  $O$ -notation of this code in terms of its argument  $n$ ? You don't have to solve the recurrence; you can just give the  $O$ -notation from your understanding of how the algorithm works.

## Problem 2 (6 points)

List two characteristics of the test-driven development process and describe (in one or two brief English sentences) why those characteristics (are claimed to) make test-driven development better than more conventional approaches.

**Problem 3** (6 points)

(a) Which of the following makes natural language understanding (spoken or written) such a difficult problem for computers? (Choose *one or more* of the following.)

- A. Individual speakers' voices vary greatly due to age, gender, native language, and regional dialect.
- B. Most modern programming languages have poorly developed facilities for string handling.
- C. Natural language is ambiguous and a lot of real-world context is required to disambiguate sentences.
- D. Continuous speech is not easily divisible into discrete individual sounds.
- E. Natural language is rich in its ability to paraphrase—there are many possible variations for expressing the same meaning.

(b) Which of the following are accurate statements about the nine different sorting algorithms shown in the film *Sorting Out Sorting*? (Choose *one or more* of the following.)

- A. The fastest algorithms sorted in  $O(n)$  time.
- B. The slowest algorithm shown was the bubble sort, which required  $O(n^3)$  time in the worst case.
- C. The film showed three groups of three sorts, ending the discussion of each group with a race among the three different algorithms. The loser of every race was an  $n^2$  algorithm.
- D. The sorts with the best  $O$ -notations were all tree-based sorts.
- E. The slowest algorithms all required  $O(n^2)$  data movements.