

**Due on Feb. 10, 2020**

**40 points total**

**General directions:**

All answers to non-programming questions must be typed, preferably using  $\text{\LaTeX}$ . If you are unfamiliar with  $\text{\LaTeX}$ , you are strongly encouraged to learn it. However, answers typed in other text processing software and properly converted to a PDF file will also be accepted. To submit your file, upload your PDF on Gradescope for assignment “HW X (PDF).” Handwritten answers or PDF files that cannot be opened will not be graded and will not receive any credit.

Finally, please read the detailed collaboration policy given on the course website. You are **not** allowed to discuss homework problems in groups of more than 3 students. **Failure to adhere to these guidelines will be promptly reported to the relevant authority without exception.**

**Point values:** Every problem has a specified amount of points which are awarded for the correctness of your solutions. In addition, each proof-oriented problem has an additional **style point**. In the homework handout, this is signified by a “+1” in the point value. To earn this point, your solutions should be clear, well organized, and easy to follow. This is to encourage not only perfectly correct solutions, but well presented ones.

**Problem 1 (9+1 points)**

Suppose  $n$  is a positive integer. Prove the following statement by using a case analysis:

For any integers  $a$  and  $b$ ,  $a - b$  is even if and only if  $a^n + b^n$  is even.

**Problem 2 (9+1 points)**

Prove the following statement using mathematical induction:

The sum of the first  $n$  positive odd integers is equal to  $n^2$ , i.e.,

$$\forall n \in \mathbb{Z}^+. 1 + 3 + \cdots + (2n - 1) = n^2.$$

**Problem 3 (18+2 points)**

Recall the two sorting algorithms: *selection sort* and *insertion sort*. These algorithms run for  $n$  rounds on an  $n$ -element array, where the  $i$ th round is defined as follows (sorting in increasing order and  $i$  starts from 1):

- **Selection sort:** Find the smallest element among positions  $i$  to  $n$  in the array, and swap it with the element at position  $i$ .
- **Insertion sort:** Insert the element at position  $i$  into the subarray from positions 1 to  $i - 1$  at the correct location, say position  $j$ , and shift all elements formerly at positions  $j$  to  $i - 1$  by one position each to their right side.

For each of these two algorithms, answer the following questions:

- (a) Can you express the property satisfied by the partially sorted array at the end of round  $i$  in each of these two algorithms? (**Hint:** *The property is different for the two algorithms.*)
- (b) For each algorithm, use its property from part (a) and the technique of mathematical induction to prove that it produces a sorted array after  $n$  rounds.