CS 8803/4803 Blockchain and Cryptocurrencies.

Solidity Programming Assignment 1

Name: Abrahim Ladha, Stan Peceny

1.1 Part 0, Setting up Truffle (0 Points)

Please see the attached `readme.txt` for instructions on how to set up Truffle. We highly recommend that you go over the CryptoZombies Tutorial.

1.2 Part 1 (80 Points)

You are provided a skeleton contract, and you are to implement the empty functions it contains.

The attached `readme.txt` contains detailed instructions and examples.

We will run 10 test cases, each is worth 8 points (5 test cases for iterative bubble sort, 5 for recursive).

1.3 Part 2 (20 Points)

Answer the following questions, please provide formal and precise explanations.

Each question is worth 2 points but question 5, which is worth 6 points.
Bonus question is worth 2 points.

Solidity

1. What version of the solidity compiler are you using with truffle? How did you determine this using truffle?
2. Explain the parameters for the first line. `pragma solidity`. What do each of the $\wedge$,<=, < operators do?
3. What is the maximum size of `uint` type in solidity? Give your answer in terms of bits. Why do you think this size is the default for `uint` in Solidity?
4. What is the difference between a `pure` and a `view` function in Solidity? Please provide and explain a use case for a `pure` function and a use case for a `view` function.
5. How much gas was used to deploy (not execute) the 1. iterative bubble sort function and the 2. recursive bubble sort function of your contract? Explain how you were able to compute this. Compare the gas used between each function and give a reason which would explain the difference.
6. Is the gas used to deploy deterministic? Explain.

7. How many local variables can you have in a Solidity function?

8. What is `msg.sender`?

(Bonus) Is Solidity Turing Complete? Please provide an in depth answer. There are a lot of wrong answers for both yes and no online, so be careful what you read.

Your deliverables include only two files. `Sorter_YOUR_NAME.sol` which is the skeleton contract provided, but fully implemented, and `Report_YOUR_NAME.pdf`, which is your solution to the questions in part 2. Please do not zip the files.