Engineering Blockchain and Web3 Apps
Problem Set #3

Problem 1. Stablecoins. During lecture we looked at how collateralized stablecoin system work. Some collateralized stablecoin system maintains collateral so that when the price of the stablecoin drops, the collateral can be used to shrink the supply of coins and bring the price back up. Some projects maintain on-chain collateral (like MakerDao, which uses ETH and other asset types for collateral) while others maintain off-chain collateral (like USDC, which uses fiat currencies such as the US dollar for collateral).

a. In MakerDao, what is the purpose of the DAI Savings Rate (DSR)? Why is it that a high DSR can be used to bring up the price of DAI, and a low DSR can be used to bring down the price of DAI?

b. In MakerDao the DSR cannot be negative. Explain what would happen if the DSR were set to -1%.

Problem 2. Oracles. In class we discussed the MakerDAO system, where DAI is intended to be a stable currency governed by MKR token holders. A brief description of the MakerDAO system is available here, and a more in-depth description is available here. Suppose that the MakerDAO pricing oracle (elected by MKR token holders) temporarily malfunctions and advertises that the price of ETH is $1,000, when in reality it is only $100.

a. How might an attacker exploit this situation to make money?

b. Assuming the error is corrected quickly enough not to destroy MakerDAO, who would bear the losses from such an attack? Describe the mechanism that causes those losses.

Problem 3. Uniswap. Recall that Uniswap uses the elegant constant product formula, $xy = k$, to determine the exchange rate between two tokens. Assuming no fees ($\phi = 1$), we showed in lecture that if the true exchange rate between two tokens A and B is M_p (i.e., $1A = M_pB$), then the market will drive the Uniswap contract to hold x tokens of type A and y tokens of type B, where $y/x = M_p$.

a. In some cases, it is beneficial to change the equilibrium point to some value other than $y/x = M_p$. To do so, suppose we change the product formula to $x^2y = k$. The market will drive this modified Uniswap contract to hold x tokens of type A and y tokens of type B, where y/x is $c \cdot M_p$ for some constant c. What is c?

b. Let us go back to the curve $xy = k$. Suppose Alice wants to buy Δx type A tokens from Uniswap. We showed in class that she would have to send $\Delta y = y \cdot \Delta x/(x - \Delta x)$ type B tokens to Uniswap. Therefore, the exchange rate Alice is getting from Uniswap is

$$\frac{\Delta y}{\Delta x} = \frac{y}{x - \Delta x}$$

.

In the open market, the exchange rate is Mp. Let us define the *slippage* s as

$$s = \frac{(\Delta y/\Delta x) - M_p}{M_p}$$

. This measures the difference in exchange rate between Uniswap and the open market (hence the name slippage). If $s = 0$ then the Uniswap exchange rate is the same as on the open market. If $s > 0$ then the Uniswap exchange rate is worse. Show that the slippage s is always positive, and is approximately $s \approx \Delta x/x$, assuming x is much larger than Δx. Use the fact that $M_p = y/x$, and that for a small $\epsilon > 0$ we have $1/(1-\epsilon) \approx 1+\epsilon$. Your derivation shows that the exchange rate in Uniswap is always worse than on the open market, however, the larger the liquidity pool, the larger x is, and therefore the smaller the slippage $\Delta x/x$, for a fixed Δx.