

# Homework 6

CS 577  
Jason Jiang

March 6, 2019

- For this problem, since there are transaction fees whenever you make USD to BDC or BDC to USD, and the transaction fees are fixed no matter how much is the transaction, we should always exchange all the currency we have to the targeted currency in one time. So, we will either have USD or BDC. Thus, for each state, the amount of currency we have is only depend on the day before the current day.

---

**Input:**  $A[1, \dots, n]$  array representing the exchange rate between USD and BDC where  $n > 0$ . Transaction fees  $b$  and  $d$ , and initial dollar amount  $k$ .

**Output:** the maximum amount of dollar at the end of  $n^{th}$  day.

**procedure** GET\_MAX\_DOLLAR( $A[1 \dots N]$ ,  $B$ ,  $D$ ,  $K$ )

$USD \leftarrow k$

$BDC \leftarrow 0$

**for**  $i = 1$  to  $n$  **do**

$USD \leftarrow \max(USD, (BDC - b) * A[i])$

$BDC \leftarrow \max(BDC, (USD - d)/A[i])$

**return**  $USD$

---

- **Proof of Complexity:**

At each step of the loop, each execution takes  $O(1)$  time, so the overall complexity of the loop is  $O(n)$ . Hence the procedure has a time complexity of  $O(n)$ .

As mentioned, the current state depends only on the last state, so it always need only 2 variables to store the results of calculation. So the space complexity is  $O(1)$ .

**Proof of Correctness:**

**Claim:** The USD is the dollar we hold on the  $i^{th}$  day and BDC is the Badger Coin we have on the  $i^{th}$  day.

**Base Case:**  $n = 1$

The max function of the USD in the loop will always return the initial USD amount if  $n = 1$  since no matter how small the positive amount we exchange the USD to BDC it will always result in less USD amount.

Since the initial BDC is 0, the max function will always pick 0 as 0 minus any positive number will result in a negative number.

Hence the return value of algorithm is correct when  $n = 1$ .

**Inductive Step:**

Assume the program is correct until the  $i^{th}$  day where USD is the current amount USD we have and BDC is the Badger Coin we have at that day.

For the  $i + 1^{th}$  day: