1. If 3-SAT $\leq_p$ BOTLOG, then BOTLOG is NP-hard.
   The expectation for this reduction is to make 3-SAT has a solution iff BOTLOG also have a solution.
   For simplicity reason I'm going to write the events like the following:
   - F(x): Fill(x)
   - E(x): Empty(x)
   - C(x, y): Check(x, y)
   - For y in check, Full will be F and Empty will be E.
   - The 3-SAT have N variables.

   Reduction:

   (a) Every variable can be represented by 2 bots and 1 box.
   For example, we have variable a. Then the bots would be:
   - $B_1 : F(1)$
   - $B_2 : E(1)$

   So now we have a assign to box 1, b assign to box 2, so on and so forth.

   (b) Let there be E number of expressions and the box to store the result start from p. So we reserved box_p to box_{p+E-1} for store the expressions result.
   For each expression$_i$ in form of $(a \vee b \vee c)$, we can create another bots to help check expression$_i$ depending on the variable (Check empty for variable with negation and Full for variable without negation). We need to use the empty box_{p+i} to help us.
   For example, we have variable $(a \vee \neg b \vee c)$. Then the bots would be:
   - $B_1 : C(1, F), F(p + i)$
   - $B_2 : C(2, E), F(p + i)$
   - $B_3 : C(3, F), F(p + i)$

   If even one of the expression is fulfill, then box_{p+i} must be full.

   (c) Lastly, we need to check whether all the expressions are fulfilled. So the bots for this step is:
   - $B_1 : C(p, F), C(p + 1, F), \ldots, C(p + E - 1, F)$

   However, the reduction above is not completed yet because we need to make sure that step (a) finish before step (b) start and also step (b) finish before we start step (c).
   So we need to create another box that to ensure that certain steps are done before executing other steps.

   Add the ordering:

   (a) Let the box to ensure the operation finish start from q.
   - $B_1 : F(1), F(q + 1)$
   - $B_2 : E(1), F(q + 2)$
   - $B_3 : F(2), F(q + 3)$
   - $B_4 : E(2), F(q + 4)$
   - $B_5 : F(3), F(q + 5)$
   - $B_6 : E(3), F(q + 6)$

   (b) In fact, we don't need to give checker from here to step (c) because of the box_{p+i} can't go from full to empty.
   - $B_1 : C(q + 1, F), C(q + 2, F), C(1, F), F(p + i)$
   - $B_2 : C(q + 3, F), C(q + 4, F), C(2, E), F(p + i)$
   - $B_3 : C(q + 5, F), C(q + 6, F), C(3, F), F(p + i)$

   (c) Because of the reason in step (b), (c) will remain unchanged.

The current reduction will still have some logs that can't be proceed even though the 3-SAT have a solution.
For example, the expression of $(a \vee \neg b \vee c)$ and we have:
$a = True$
$b = True$
$c = True$

B_2 in step(b) won't finish even though the it doesn't matter anymore because the other 2 variables already make the expression true.
So we need to flush all the remaining logs. So we need another bot where we are sure that all the expressions are true and change all the variable possibilities. The bot is:
- $B_1 : C(p, F), C(p+1, F), \ldots, C(p+E-1, F), F(1), F(2), \ldots, F(N), E(1), E(2), \ldots, E(N)$

So now we have several steps and bots:

(a) - $B_1 : F(1), F(q+1)$
- $B_2 : E(1), F(q+2)$
- $B_3 : F(2), F(q+3)$
- $B_4 : E(2), F(q+4)$
- $B_5 : F(3), F(q+5)$
- $B_6 : E(3), F(q+6)$
- etc.
- $B_{2*N-1} : F(N), F(q+2*N-1)$
- $B_{2*N} : F(N), F(q+2*N)$
- $O(2 \cdot 2 \cdot N) = O(N)$

(b) Let $k_1 = 2 * N$
- $B_{k_1+1} : C(q+1, F), C(q+2, F), C(1, F), F(p+i)$
- $B_{k_1+2} : C(q+3, F), C(q+4, F), C(2, E), F(p+i)$
- $B_{k_1+3} : C(q+5, F), C(q+6, F), C(3, F), F(p+i)$
- etc.
- $O(4 \cdot 3 \cdot E) = O(E)$

(c) Let $k_2 = k_1 + 3 \cdot E$
- $B_{k_2+1} : C(p, F), C(p+1, F), \ldots, C(p+E-1, F), F(1), F(2), \ldots, F(N), E(1), E(2), \ldots, E(N)$
- $O(E + 2N) = O(E + N)$

So the transformation cost is $O(N + E)$ which is in polynomial time.

Let Φ be our 3-SAT

Claim: If the 3-SAT is satisfiable, then BOTLOG has a solution.
Proof:
Suppose the 3-SAT has satisfying assignment x^*. Then we ordered the bot as follows:

(a) If $x_i^* = 1$ then we put B_{2*i} logs before B_{2*i-1}.
If $x_i^* = 0$ then we put B_{2*i} logs after B_{2*i-1}.
Now every x^* is represented by box_i

(b) For every expression$_y$ that takes x_i, x_j, x_k as it's variable has 3 bots and won't be able to run before step (a) of their part is done because they need to ensure that $C(q+2 \cdot i, F)$ *and* $C(q+2 \cdot i-1, F)$. If any one of the variable is correct, it'll make the bots to $F(p+y)$.

(c) The last bot is checking whether all the expressions are all $True$. If yes, then it can clear all the remaining logs in step(b).

Claim: If BOTLOG has a solution, then 3-SAT is satisfiable.
Proof:
- The solution of the satisfying assignment x^* should be depending on which bot goes later. x_i^* is depend to box_i. If $B_{2 \cdot i-1}$ goes later than $B_{2 \cdot i}$, box_i is Full. And vice versa.
- If the bots has make box_{p+i} full, then it means that expression_i is $True$. The bots has already done one of the 3 box checking.
- The rest of the logs doesn't matter to the 3-SAT result since it's only cleaning the remaining steps that is undone in step (b).

2. The certificate is the BOTLOG solution. The solution provide which bot do each command and what command it does. Several steps to check the solution:

   (a) Check the length of the solution match to the sum of all logs in all bots. Iterate through all bots and solution will take $O(E)$.

   (b) Check the solution whether it contradict with itself.
   This can be done by running the solution and keep track for every box. If there exist even one contradiction, the certificate is false. This part runs in $O(N)$.

   (c) Check the solution whether it contradict with the bots.
   Give pointer that start at command_1 for each bots. And iterate through the solution. For example, the command is:
   Bot 4: Fill(3)
   So the if the pointer not pointing to Fill(3) in Bot 4, then the certificate is false. Else, move the pointer to the next log in bot4.
   This part will runs in $O(E)$.

   Total running time is $O(E + N)$ which is in polynomial time.
   Since the certificate is the solution. Then if it's a valid bot log then there exist a solution. Because there exist a solution, then the certificate also exist.